DOCUMENT RESUME

ED 213 027 CS 206 743

AUTHOR	Holland, V. Melissa; Rose, Andrew
TITLE	A Comparison of Prose and Algorithms for Presenting Complex Instructions. Document Design Project, Technical Report No. 17.
INSTITUTION	American Institutes for Research, Washington, D.C.; Carnegie-Mellon Univ., Pittsburgh, Pa.; Siegel & Gale, Inc., New York, N.Y.
SPONS AGENCY	National Inst. of Education (ED), Washington, DC. Teaching and Learning Program.
REPORT NO	AIR-75003-11/81-TR
PUB DATE	Nov 81
CONTRACT	400-78-0043
NOTE	90p.
EDRS PRICE	MF01/PC04 Plus Postage.
DESCRIPTORS	*Algorithms; College Students; *Comparative Analysis; *Government Publications; *Layout (Publications); Problem Solving; *Prose; Reading Difficulties; *Reading Research; Sentence Structure
IDENTIFIERS	Direction Following

ABSTRACT

Complex conditional instructions ("if X, then do Y") are prevalent in public documents, where they typically appear in prose form. Results of two previous studies have shown that conditional instructions become very difficult to process as the structure become more complex. A study was designed to investigate whether this difficulty can be alleviated by presenting conditional instructions in formats other than prose. It was hypothesized that the major sources of difficulty--disjunction, negation, hierarchical structure, and ambiguous punctuation--would be eliminated by the use of algorithms. Subjects were presented with a sample of complex conditional instructions in three formats--prose and two forms of algorithms, lists and flow charts. Results showed that conditional instructions were easier to follow when presented as algorithms than as prose when subjects were prepared to follow algorithms by practice with feedback. In general, the algorithm helped performance when the condition in the instruction was logically complex. The overall superiority of algorithms was qualified, however, by three complications: (1) the difference between flowcharts and lists changed the speed relationships between prose and algorithms, (2) the effect of partial processing changed the speech relationships, and (3) the difficulty with the first exposure to the instruction drastically reversed the algorithm advantage in both speed and accuracy. (HOD)

ED213027

AIR-75003-11/81-TR

DOCUMENT DESIGN PROJECT

Technical Report No. 17

A Comparison of Prose and Algorithms for Presenting Complex Instructions

V. Melissa Holland
Andrew Rose

A product of the Document Design Project funded by the National Institute of Education, Washington, D.C
Contract No. NIE-400-78-0043

November 1981

AMERICAN INSTITUTES FOR RESEARCH/1055 Thomas Jefferson Street, NW, Washington, DC 20007 • 202/342 5000
with Carnegie Mellon University and Siegel & Gale Inc

CS 206 743

ACKNOWLEDGEMENTS

We are indebted to several people at AIR who helped us with this paper and with the research reported here. Don McLaughlin, Laurie Wise, and Paul Fingerman wrote the computer program we used to randomize and present the experimental stimuli and to score the responses. Laura Malakoff and Debbie Leinbach assisted as experimenters for the study. Don McLaughlin and Ron Harnar helped with the computer analyses of the data. George Wheaton advised us in interpreting the results of the analyses. Ginny Redish provided editorial comments on the written report, and Mary Medved oversaw the typing and formatting of the report.

This report is part of the work done under contract #400-78-0043 of the National Institute of Education. The project officer is Dr. Candace Miyamura of the Teaching and Learning/ Reading and Language Group. This report does not necessarily reflect the views of the funding agency.

ABOUT THE DOCUMENT DESIGN PROJECT

In September 1978, the American Institutes for Research (AIR) began the Document Design Project to foster clear and simple writing and design of public documents. The purpose of the Document Design Project (DDP) was to help make forms, regulations, brochures, and other written materials easier for people to read, to understand, and to use. Carnegie-Mellon University and Siegel & Gale, Inc. worked with AIR on this project. The project was funded by the Teaching and Learning/Reading and Language group at the National Institute of Education.

The Project's goal was to increase the knowledge and skills of people who produce public documents. To accomplish this goal, the Document Design Project had three tasks:

> Task 1: To conduct theoretical and applied research studies on language comprehension, on the ways in which skilled and unskilled writers work, on problems associated with different document features;
>
> Task 2: To bring research into practice by working with government agencies as they produce materials for public use;
>
> Task 3: To bring research and practice into education by developing courses on writing and design for graduate students and undergraduates.

If you have questions or comments on this report or on other work of the Document Design Project, contact Dr. Janice C. Redish, Director, The Document Design Center, AIR, 202/342-5071.

TABLE OF CONTENTS

A Comparison of Prose and Algorithms for Presenting Complex Instructions

I. INTRODUCTION

This paper reports on the third in a series of experiments on how people follow instructions with complex conditions. These instructions are conditional sentences of the form: "If X, [then] do Y," in which the antecedent ("if") clause includes several categories linked coordinately.* An example is:

> "If you are a parent or a homeowner, and not both under 26 and a veteran, mark Box A."

In this example, "parent," "homeowner," "under 26," and "veteran," are the categories of the antecedent. The coordinate structure linking the categories is specified by various combinations of the connectives "and," "or," and "not." These are the basic connectives that link propositions in truth-functional logic.

Complex conditional instructions are prevalent in public documents, where they typically appear in prose form, like the sentence above. In the two previous experiments (Rose & Cox, 1980; Holland & Rose, 1980), we investigated the speed and accuracy of subjects' responses to these instructions when we manipulated the logical structure of the antecedent. In those experiments, the goal was to identify structural determinants of processing difficulty. The variables of interest lay in the logical connectives between the content categories rather than in the categories themselves.

The results of these experiments showed that conditional instructions become very difficult to process as the structure of the antecedent becomes more complex. The current study was designed to investigate whether this difficulty can be alleviated by presenting conditional instructions in formats other than prose. Specifically, the study addressed five questions:

- Given the coordinate structures identified as difficult in the previous experiments, could we facilitate performance by presenting the conditional instruction as an algorithm?

*A sentence in "if-then" form with an imperative clause "do Y" in place of the consequent ("then Y") does not fall into the taxonomy of true conditional sentences defined by logicians (Quine, 1972). Failing to find an a priori niche, we have decided to call these expressions "conditional instructions."

- What kinds of logical or syntactic difficulties in prose processing does an algorithm alleviate?

- How does performance compare between two forms of algorithms: (1) flowcharts and (2) lists of steps ("jump" questions)?

- How does a reader's performance on first exposure to algorithms compare to performance on prose instructions?

- What format does the average reader prefer for complex conditional instructions?

Our goal is a practical one: We want to know what to tell a writer to do when the material includes complex conditional instructions. But we also want to understand what readers are doing. Therefore, the preceding studies, in addition to identifying structural correlates of processing difficulty, explored the theoretical questions of the nature and causes of this difficulty. We explained observed performance by relating logical and syntactic description to inferred cognitive operations in a set of processing models. These models were essentially task analyses that took into account both comprehension (decoding and representing the structure of the antecedent) and use (evaluating the truth of the antecedent to decide on a response to the instruction).

We built our hypotheses for the current study on our theoretical explanation of the difficulties found in the earlier experiments. According to our task analyses, the density of the internal cognitive operations required to represent and use the antecedent in an instructional task determines the level of difficulty of the conditional instruction. Simply on analytical grounds, the algorithm can be predicted to facilitate processing, because it breaks problem-solving into a set of discrete, linear, and external steps. Figure 1 on page 3 illustrates how this break-down occurs in a flowchart algorithm applied to the prose sentence presented on page 1. Figure 2 on page 3 shows a list algorithm for the same sentence.

Beyond analytical grounds, we lacked empirical bases for deriving predictions about the effects of algorithms on processing coordinate logic. In a review of the empirical literature on human performance with algorithms (Holland, 1981), we observed that the accumulated research is largely ad hoc and ungeneralizable. Although a few recent studies have experimentally compared human performance with algorithms to performance with prose (for example, Wason, 1968; Jones, 1968; Davies, 1970; Wright & Reid, 1973; Blaiwes, 1974; Wright, 1975; Kammann, 1975; Landa, 1976; Follettie, 1979), many of them are simply demonstrations of the advantages of using an algorithm.

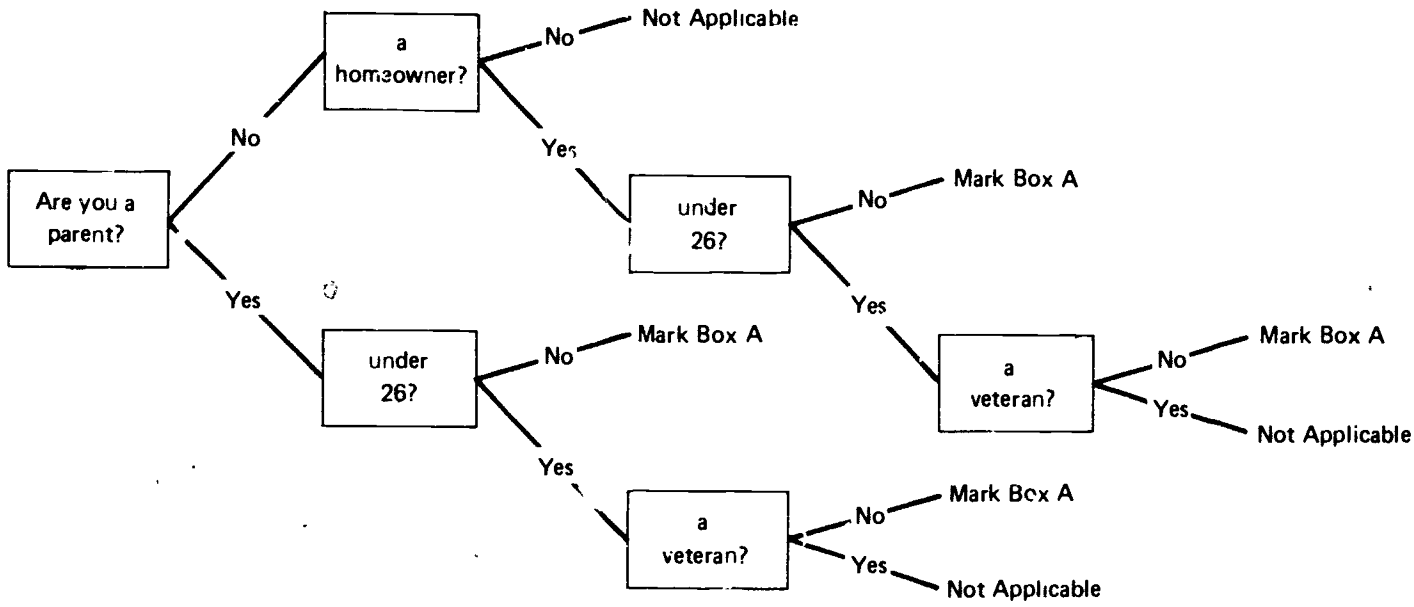


FIGURE 1. Flowchart algorithm for the instruction, "If you are a parent or a homeowner, and not both under 26 or a veteran, mark Box A."

(1)	Are you a parent?	If <u>yes</u>, go to 3. If <u>no</u>, go to 2.
(2)	Are you a homeowner?	If <u>yes</u>, go to 3. If <u>no</u>, go to 5.
(3)	Are you under 26?	If <u>yes</u>, go to 4. If <u>no</u>, go to 5.
(4)	Are you a veteran?	If <u>yes</u>, go to 6. If no, go to 5.
(5)	Mark Box A.	
(6)	Not applicable.	

FIGURE 2. List algorithm version of instruction, "If you are a parent or a homeowner, and not both under 26 or a veteran, mark Box A."

These studies characteristically fail to define (or lack a systematic approach to defining) the variables that are critical to performance--the functional features of the task, the graphic features of the format, and the structural features of the instruction.* The present study is the first attempt we know of to compare prose with alternative instructional formats while varying the logical structure of the instruction in precise and systematic ways.

We will begin our description of this study by reviewing the results from the previous study of prose instructions. We will then analyze the cognitive tasks required by prose and compare them with the tasks required by algorithm formats for these instructions. Based on these analyses, we will describe the expected results for each of the five questions posed at the beginning of this section.

Difficulties with Prose: Previous Results

The prose sentences used in the previous experiment (Holland & Rose, 1980) selectively sampled the universe of compound coordinate structures. We can think of this universe as generated by the propositional calculus of truth-functional logic, in which propositions may be conjoined by the operations of conjunction ("and"), disjunction ("or"), and negation. To construct the sample, we varied the stimuli on four dimensions. These were:

- the number of categories in the compound--four, five, six, or seven;
- the connective between categories--"and" vs. "or";
- affirmative vs. negative categories (simple negation)--no "nots" vs. two "nots" ("not A and not B");
- The external organization of the categories--
  (1) strings ("A, B, C, and D"), that is, linear structures;
  (2) simple groupings ("A or B, and C or D"), that is, hierarchical structures;

*See Follettie (1978) for a critical analysis of the methodology of several of these studies.

(3) groupings ("A, B, and neither C nor D") built around compound negation.*

The structures created by the factorial combination of these dimensions were placed in sentence frames of the form: "If you are ________, press button A(B)." The example,

"If you are male and employed, or single and over 25, press button A,"

is a structure of the type "four-category, 'or,' affirmative, simple-grouping." We classify this as an "or" sentence because the connective between the two major groups is "or." (An "and" grouping would have the form "A or B, and C or D.")

We used high-frequency words and phrases, like "male" and "over 25," to fill the category slots in the coordinate structures. Subjects were told to apply these categories to themselves following the underlying structure of the "if" clause, and to then respond on the basis of the overall truth or falsity of the clause. We measured the logical difficulty of a structure in terms of both the speed and the accuracy of subjects' responses to the instructions.

From the results, we were able to identify significant aspects of logical difficulty in terms of the four dimensions manipulated in the stimulus sentences:**

- Groupings were, on the whole, harder than strings.

- Negative groupings were the hardest organizational type. We attributed this to the difficulty of joint and alternative denial ("not both/and," "not either/or," respectively).

*Note that the third dimension (affirmative vs. negative category) refers to simple negation ("not A")--modifying a single category rather than a group. But under the fourth dimension (organization), compound negation ("neither A nor B") refers to negating groups. Linguistically, this is a difference in scope of the negative term. Logically, compound negation is referred to as joint denial and alternative denial. Previous research has demonstrated the difficulty of those two logical operations (e.g, Haygood & Bourne, 1965).
**Various of the significant factors identified in this experiment were found in Rose and Cox (1980), using two-category conditional instructions, and have also been isolated in other contexts--such as in psycholinguistic studies of syntactic effects and concept-learning studies of logical rules. These studies are reviewed in Holland and Rose (1980).

- Structures with negative categories were harder than structures with no negatives.

- "Or" strings ("A, B, C, or D") were harder than "and" strings ("A, B, C, and D.").

- Simple groupings in which "and" links the major groups ("A or B, and C or D") were harder than the reverse ("A and B, or C and D"). We attributed this result to the fact that the "and" structure has more "ors" (as subordinate connectives) than does the "or" structure.

- Any structure with "ors" in a series of three or more ("A, B, or C; and D, E, or F") was harder than paired "ors" ("A or B, and C or D, and E or F"). We attributed this result to the potential ambiguity of "ors" expressed implicitly in commas.

Figure 3 shows more clearly the order of difficulty of the structures of interest from the previous experiment. Although we were generally able to make meaningful statements about difficulty only in terms of interactions of factors, we could nevertheless distill three components of logical difficulty from the obtained ordering. These components were negation, disjunction ("or"), and hierarchical organization of categories (groupings).

Hard ⟶ Easy

structure:	"not" grouping	"and" grouping	"or" string	"or" grouping	"and" string
example:	A, B, or not both C and D	A or B, and C or D	A, B, C, or D	A and B, or C and D	A,B, C, and D

Figure 3. Order of difficulty of structures from Holland and Rose, 1980, (based on accuracy and response time data combined).

The number of categories by itself did not affect errors. Number did, on the other hand, add a constant time increment, since subjects consistently processed the full instruction instead of selecting the relevant parts of it. Partial processing was possible for most instructions. For example, someone reading the instruction, "If you are A or B or C, press X" can skip to the imperative clause after reading the first category ("A") if it is true.

In addition to the structural variables already mentioned, the previous experiment also varied the surface expression of the connective in the conditional clause. Signifying "or" implicitly with commas ("A, B, C, or D") was found to be harder than repeating it ("A or B or C or D"). Commas apparently create ambiguity in the "or" structure because they tend to be read as "ands."

The current study follows directly from the previous experiment in two ways. The stimuli here were modeled on the prose sentences used in the previous experiment. T : current hypotheses about how and where algorithms will improve performance were based on the patterns of difficulty observed in the prose sentences.

Task Analyses for Prose and Algorithms

Our broad hypothesis is that the major sources of difficulty in complex conditional instructions--disjunction, negation, hierarchical structure, and ambiguous punctuation--will be eliminated by the use of algorithms. This hypothesis comes from understanding how the prose instructions are processed, and from comparing how the same instructions are processed as algorithms. Let us analyze the prose task first and then the algorithm task.

How we process complex conditional instructions as prose

Subjects in the previous study were required to read and follow instructions of the form, "If you are [coordinate compound], press button X (otherwise, press Z)." We analyzed this task into three theoretical steps:

(1) Interpreting the coordinate compound in the "if" clause--i.e., extracting the logical rule. (This step might be divided into "reading" and "rule extraction/construction," with reading taking the smaller amount of effort.)

(2) Applying the compound condition, once interpreted, to oneself--i.e., verifying the clause.

(3) Using the verification decision (was the condition true or false?) to determine a response to the imperative clause (whether to "press X" or not).

It is steps (1) and (2)--extracting and applying the logical rule--that created most of the problems in the previous experiment. What is required in these processes?

First, extracting the rule requires recovering the logical structure of the compound in the conditional clause. This structure is defined by the connectives ("and," "or," "not") and the way they are grouped. Recovering the logical structure entails mentally constructing a psychological "truth table" (or decision table). This table shows how the truth values of the individual categories constituting the compound must be combined to yield an overall "true" or "false" outcome. There may be several different sets of values which yield a true or a false outcome. Figure 4 illustrates truth tables for three sample sentences. A partial truth table, as in Figure 4, gives only the "true" or only the "false" outcomes.

A and B and C

	T
A	T
B	T
C	T

A or B or C

	F
A	F
B	F
C	F

A and B, or C

	T	T	T	T	T
A	T	T	F	F	T
B	T	T	F	T	F
C	T	F	T	T	T

Figure 4. Partial truth tables for three sample sentences.

The next step, applying the rule, entails two tasks: one task is making the successive binary decisions--true or false--about the individual categories; another is assessing the overall truth value of the condition. Making this assessment requires comparing the pattern of one's own truth values for the individual categories against the patterns in the psychological truth table to determine a match.

Given this general explanation of how conditional instructions in prose are processed, how do we explain the particular patterns of errors obtained? We can infer that the easier sentences are easier because the decision rule is simpler. For example, for "and" strings the rule is: "at least one 'no'

means false, all 'yesses' means true." The simplicity of this rule is reflected in the truth table. In the table for "and" strings, only one column--a single combination of category values--is necessary to specify the set of "true" outcomes. On the other hand, groupings of any sort, which are more difficult to process, require at least two columns in the truth table (as in Figure 4). Thus, the number of columns required in a partial truth table to specify all possible "true" combinations of categories (or, alternatively, all possible "false" combinations) appears to be a metric of the underlying logical complexity of coordinate structures.*

How we process conditional instructions as algorithms

To compare the cognitive requirements of prose to those of algorithms for presenting complex conditional instructions, let us analyze the task of using the algorithm in general, extracting those properties with consequences for processing coordinate logic. We will focus on that core of properties that is common to flowchart and list algorithms.**

Algorithms are procedures for solving a problem that break the process into its simplest steps and arrange them in a sequence leading automatically to the right outcome. Figures 1 and 2 (p. 3) illustrate how the algorithm breaks a continuous prose instruction into a sequence of simple "yes-no" questions and connects them with final outcomes (e.g., "Mark Box A") in ways that reflect the underlying logical contingencies. Note that only the <u>minimum number</u> of <u>relevant categories</u> for a given contingency is connected to an outcome. For example, for two combinations of truth values ("not a parent, not a homeowner," and "a parent, not under 26"), readers exit after reading only two of the four categories of the instruction shown in Figure 1. The minimum number of relevant categories varies with the nature of the logical structure and the configuration of truth values for a given reader.

*According to this criterion, however, "and" and "or" should be equivalent in complexity. There is no formal correlate of the "and-or" asymmetry so prevalent in our results, since the respective truth tables are mirror images of each other. That is, a partial truth table for "or" could be constructed as in Figure 4, containing a single column: all "noes" producing a "false" outcome. There is no basis in formal reasoning for why "noes" and "falses" should count differently from "yesses" and "trues." We must rely instead on psychological criteria, such as negative signs being more complex, to explain why "or" sentences are harder than "and."

**Some of these properties have been discussed elsewhere (e.g., Gane, Horabin, & Lewis, 1966; Wheatly & Unwin, 1972; Wason & Johnson-Laird, 1972) and have some empirical support.

To use the algorithm, readers must read the first question, select the appropriate answer, then read the question or command immediately connected to that answer. The path is indicated by arrows (the flowchart) or "go to" commands (the list). Thus readers follow only one path through the algorithm, contingent on their answers at each previous step. The memory operations required to move through the sequence span a single step: performance does not require rule extraction or construction of a psychological truth table.

Given this analysis of the task, there are four properties that suggest an advantage for algorithms over prose instructions:

- Algorithms reduce syntactic and logical structure to a series of binary decisions. In this way, algorithms eliminate the words ("if," "otherwise," "but," "not," "and") and the grammatical devices (coordination, embedding) required to express relationships and to connect them with contingent decisions.

- Algorithms clarify ambiguous punctuation and syntax.

- Algorithms relieve the burden of remembering a train of antecedent decisions or of back-tracking to recover them.

- Algorithms build in partial processing: they lead you to read only what is necessary to reach an outcome.

These four properties should eliminate the main sources of processing difficulty we have isolated for prose. Specifically, algorithms should:

- relieve readers from extracting and applying the logical rule underlying the conditional clause,

- eliminate the possibility of misinterpreting ambiguous commas in "or" strings,

- relieve readers from keeping track of previous truth values, and

- prevent processing of categories irrelevant to a given outcome.

Together, these inherent simplifications of the instructional task lead us to predict that algorithms will alleviate the difficulty of processing coordinate logic that we observed with prose instructions.

Drawbacks of Algorithms. So far we have focused on the potential advantages of algorithms in performance. We must also ask whether any properties of algorithms suggest disadvantages relative to prose. Three such properties emerge from analytic considerations:

- Algorithms require an overt response to each binary decision. Prose allows individual binary decisions to occur internally and thus more quickly.

- Algorithms separate the component questions physically and graphically to a greater degree than does prose, requiring more time for readers to move between categories.

- Algorithms are less familiar to look at and to operate with then is prose for most readers.

From these properties, we might expect that instructions with easier logic will be slowed down by the algorithm in cases where they require full processing (or where the reader is already a partial processor of prose). Further, because it is unfamiliar, the new format may confuse readers.

Another property of algorithms has a less predictable effect. Algorithms afford no insight into the structure of the rule. Prose, on the other hand, requires readers to construct the rule themselves and thus grasp the structure. Readers may not like the algorithm because it is "blind." This attitude in turn may impede performance or discourage use.*

Lists vs. flowcharts

The third question posed in this study concerns the relative benefits of different forms of algorithms. Given our expectations of improved performance with algorithms generally, how will lists and flowcharts compare in time and errors?

Figure 2 depicted the list algorithm as a series of "jump" questions (in the terms of Wason, 1968) that reflect the structure of the flowchart. The component questions are presented as items

*For functions other than presenting instructions--e.g., algorithms for learning and remembering--the lack of insight in the algorithm is more clearly a potential handicap (as suggested in Holland, forthcoming). We should point out that, beyond our particular experimental task and instructions, it is the function intended and the type of material to be formatted that determine what the inherent properties of algorithms will mean: in some cases enhancing, in others, diminishing performance.

in a list rather than as signs in boxes. The "go to" commands in the list specify "jumps" that allow the reader to skip over irrelevant questions and that capture the logical contingencies between questions and outcomes.

We have little to go on in formulating hypotheses about the relative benefits of the list and the flowchart. There is no previous research comparing the two forms of algorithms. But from the appearance of the two sample formats (Figures 1 and 2), we can analyze differences in the task demands and draw implications for performance.

Formally, the flowchart and the list of questions are the same. (To derive one from another, you may simply exchange questions for flowchart boxes, and "go to" commands for arrows.) Both forms translate the categories of prose into simple questions. Both direct the reader through a succession of these questions, touching only the relevant ones, contingent on the reader's responses at each step.

But the two formats differ in the detailed requirements of both design and use. The flowchart as described here repeats an outcome at each possible exit point in the route through the questions.* The list, like prose, states the outcomes once, at the end of the questions. Thus, the flowchart typically represents branched structures by duplicating question boxes (Figure 1) while the list, like prose, states each question once (Figure 2). Finally, the list uses prose statements rather than graphic symbols to guide the reader. Where the list <u>tells</u>, the flowchart <u>shows</u> the reader where to go.

These features imply that flowcharts may be simpler in terms of ideal information processing: The flowchart directs the reader visually with arrows and duplicates boxes and outcomes. This feature eliminates the possibility of increased time and error in "jumping" between questions, or between questions and outcomes, in the list. The list requires the extra operations of encoding and remembering the item numbers specified in "go to" commands and concurrently scanning the array of questions in search of a match.

Note, however, that the list has certain psychological and design advantages following from the fact that it is closer to prose than is the flowchart:

- The list is generally more efficient than the flowchart in that it takes up less space on the page and does not repeat questions and outcomes.

*Of course, we could have constructed the flowchart in Figure 1 without duplicating questions or outcomes by using recursive and cross-over arrows. (See p. 21)

- The list may be more acceptable in that it is more familiar and less technical looking than the flowchart.

- The list is easier to interpret for the naive reader who does not know what arrows and boxes mean.

Nevertheless, our basic expectation is that the list format will be somewhat more difficult than the flowchart in an experimental setting because of the extra scanning and memory operations involved.

First exposures

Another of our motivating questions was how readers will respond the first time they see an algorithm in a given context. The question is crucial to the practical aims of this study. Readers who encounter a flowchart on a tax form or other public document are not going to receive operating directions and practice trials before having to deal with the flowchart. The document designers to whom we would make format recommendations care about the effect of first exposures.

Our observation that algorithms are less familiar than prose to the average reader suggests that readers may perform worse with the algorithm than with prose in their first exposure to a set of instructions.

Format preferences

Another consideration in making recommendations about formats is how readily people will accept algorithms in real-world documents. Thus the final question in this study concerned readers' format preferences. Our earlier description of algorithms as unfamiliar and automatic is relevant here. These characteristics suggest that people may not like algorithms as well as prose.

Prior evidence about attitudes toward and acceptance of new formats is scant and often limited to special populations (e.g., Kammann's survey of Bell Labs engineers, 1975). Inhibiting attitudes are likely to be tied to levels of education and experience in the population at large. That is, people who rarely use flowcharts or lists may not readily accept them.

General Experimental Approach

To test the questions discussed above, we used the following approach: In the main condition, we presented subjects with a sample of complex conditional instructions in three formats--prose and the two forms of algorithms. We measured response time and errors after providing practice and feedback in each format, and ended the session with questions about preference and experience. We prefaced the main experiment with a set of "cold" trials, presented before practice and feedback. The set consisted of four trials in each format. This "unpracticed condition" served primarily to test the effect of first exposures on performance. It also allowed us to look at an important structural variation that would have over-burdened the main design--simple affirmative vs. negative relations. We followed the test session with questions about subjects' format preferences and prior experience with algorithms.

II. STIMULUS DESCRIPTION

All stimuli were conditional instructions of the basic form

If you are X, press [key]Y

where X was a coordinate compound of several categories (e.g., "A or B, and C or D"), and "press Y" was a simple instructional command contingent on the truth value of the compound.

Stimuli varied on six dimensions. These were:

(1) The format of instructional presentation:
- prose;
- flowchart algorithm;
- list algorithm

(2) The number of categories contained in the coordinate compound:
- four categories
- six categories

(3) The type of connective between categories:
- conjunctive ("and")
- disjunctive ("or")

(4) The organization of categories--whether the were grouped, and, if so, how:
- strings ("A, B, C, and D")
- simple groupings ("A or B, and C or D")
- "not" groupings ("A, B, and not either C or D")

(5) Truth value of the instruction:
- true
- false

(6) "Exit position": the point in the instruction where an ideal reader could correctly make a key press response:
- early exit
- middle exit
- late exit

Another possible dimension, "affirmative-negative," was used in the previous experiment. In the present case, we decided to include two negated categories in each instruction in the main experiment.*

*We chose the negative value of this dimension because our interest was in comparing formats for the instructions which proved most difficult in the previous experiment.

The last two dimensions--truth value and exit position--require some further elaboration. They served primarily as control variables on processing time and difficulty. They are not strictly structural since both depend on the content of the categories in the compound (the words themselves)--or, more precisely, on the interaction of content with the characteristics of the reader. We will illustrate these two dimensions using the two-category conditional instruction, "If you are male or married, press button Y; if not, press Z."

The overall truth value of the condition, "you are male or married," is "false" for a single female, "true" for any male as well as for any married person. Thus there is more than one pattern of individual truth values that yields a "true" outcome. The patterns are illustrated in truth table fashion, in Table 1.

Truth Values for Individual Categories		Overall Truth Values
male	married	male or married
T	F	True
F	T	
T	T	
F	F	False

Table 1. Truth Table for "X is male or married."

The pattern of truth values for the individual categories determines the possible exit points in processing. Assuming a left-to-right order in processing categories, the TF and the TT patterns allow an "ideal" male reader to decide the overall truth value of the condition after evaluating only the first category ("male")--given that the read r has observed the "or" coordinate structure. The reader may then "exit" and go to the imperative clause--that is, press Y. This would count as an early exit or partial processing. On the other hand, the FF and FT patterns require full processing, or "late" exits.

As in this example, there is for any logical compound always more than one way to combine and sequence the individual values to yield at least one of the overall truth outcomes (True or False). The location of the individual trues or falses in the sequence of categories in a structure determines when a particular reader can decide the overall outcome and exit from a structure; or, more precisely, how many categories the reader must process before exiting. That the distribution of truth values to yield a given exit position is structurally dependent can be illustrated by comparing a linear organization with a grouping. The earliest exit from an "A, B, C, and D" string is a pattern with a false value for the first category: In the shortest case, reading

terminates after a single category is processed. By contrast, the earliest exit for the "A or B, and C or D" grouping is a pattern with "false" values for A and B or with "true" values for A and C: In the shortest cases, reading terminates after two categories are processed.

For any four- or six-category structure, there are several choices of when (after how many categories are read) an exit can occur. And for each exit position specified, there are usually several determining distributions of "true" and "false" categories. We chose three levels for the exit position variable, which we called early, middle, and late exits. The meaning of these levels depends in part on the organizational structure of the compound, as demonstrated above; in part on the length of the compound; and in part on arbitrary definition.

Table 2 on the next page defines the early-middle-late variable for each of the 12 truth-functional structures, crossed with the two possible truth values for each structure. The definition of each level of the variable is in terms of the number of minimally relevant categories in the condition (the number that must be evaluated before a verification decision can be made on the condition as a whole). The "early" exits were always defined as the least possible processing allowed by the structure and truth value in question. The "late" exits were defined as full processing--i.e., requiring evaluation of all categories in the sequence. There was more freedom in defining the "middle" exits, and we somewhat arbitrarily set these at three categories to be evaluated in four-category structures, and four categories in the six-category structures.*

For some structures, it is inherently impossible to distinguish three exit positions for both "true" and "false" outcomes. This is clear from the string structures in Table 2. For "and" strings, the only way to construct a "true" condition is by making all categories false (with reverse transformations to account for the two negated categories that characterize each structure in our sample). Thus full processing is logically necessary to recognize false conjunctions. In cases such as this, we simply replicated the earliest level (or levels) of exit position that was available and used those replications for the exit-position variable.

Since the average number of categories to be processed in four- and six-category instructions varies across organizational type, the "early-middle-late" dimension does not have equal intervals, as shown in Table 2. Table 2 also shows, beside each specification of the number of categories to be processed, the particular pattern of truth values selected to generate that exit level. Each pattern was selected at random from the set of all

*There was one exception, for the "not" grouping, as indicated in Table 2.

TABLE 2.
The Stimulus Set

STRUCTURE					TRUTH VALUES			
					True		False	
Organization	Number of Categories	Connective	(Example) [a]	Exit Position [b]	Exit Number [c]	Truth Pattern [d]	Exit Number [c]	Truth Pattern [d]
Strings	4	And	(A, B, $\overline{C}$, & $\overline{D}$)	E	4	T T F F	1	F T F F
				M	4	T T F F	3	F F F T
				L	4	T T F F	4	F F T F
		Or	(A, B, $\overline{C}$, or $\overline{D}$)	E	1	T F T T	4	F F T T
				M	3	T T T F	4	F F T T
				L	4	T T F T	4	F F T T
	6	And	(A, B, C, D, $\overline{E}$, & $\overline{F}$)	E	6	T T T T F F	1	F T T T T T
				M	6	T T T T F F	4	T T T F F F
				L	6	T T T T F F	6	T T T T F T
		Or	(A, B, C, D, $\overline{E}$, or $\overline{F}$)	E	1	T F F F F F	6	F F F F T T
				M	4	F F F T T T	6	F F F F T T
				L	6	F F F F T F	6	F F F F T T
Single Groupings	4	And	(A or B, & $\overline{C}$ or $\overline{D}$)	E	2	T F F T	2	F F F F
				M	3	T F T F	3	T T T T
				L	4	F T T F	4	F T T T
		Or	(A & B, or $\overline{C}$ & $\overline{D}$)	M	2	T T T T	2	F T T F
				M	3	F F F F	3	F T F T
				L	4	T F F F	4	T F F T
	6	And	(A, B, or C; & D, $\overline{E}$, or $\overline{F}$)	E	2	T F F T T T	4	T T T F T T
				M	4	F T F F F T	4	T T T F T T
				L	6	F F T F T F	6	F F F F T T
		Or	(A, B, & C, or D, $\overline{E}$, & $\overline{F}$)	E	4	F F F T F F	2	F T T F F F
				M	4	F F F T F F	4	T F T T T F
				L	6	T T T T F F	6	T T F T F T
"Not" Groupings	4	And	($\overline{A}$, $\overline{B}$, & not either C or D)	E	4	F F F F	1	T F F F
				M	4	F F F F	3	F F T F
				L	4	F F F F	4	T T T F
		Or	($\overline{A}$, $\overline{B}$, or not both C & D)	E	1	F T T T	4	T F F F
				M	3	T T F T	4	F F T F
				L	4	F F F T	4	T T T F
	6	And	(A or B, & not either C or D, & $\overline{E}$ or $\overline{F}$)	E	4	T F F F F T	2	T T T F F F
				M	4	T F F F F T	4	F T T F T T
				L	6	F T F F T F	6	F T F F T T
		Or	(A & B, or not both C & D, or $\overline{E}$ & $\overline{F}$)	E	2	F F F T T T	4	F T T T T F
				M	3	T F F T F F	4	F T T T T F
				L	6	T F T T F F	6	T F T T F T

[a] "$\overline{A}$" refers to "not A," "&" refers to "and."

[b] E, M, L = Early, Middle, Late exits

[c] Given as the number of minimally relevant categories in a particular structure-truth value combination

[d] Given as the sequence of truth values corresponding to the individual categories in the condition

possible patterns determining a specific exit position for a particular structure-truth value combination.

In summary, the complete stimulus set for the main condition consisted of all combinations of three "formats," two "numbers," two "connectives," and three "organizations," yielding 36 unique stimuli. Each of these 36 could be either true or false, and each could have an early, middle, or late exit. Thus, a total of 216 (i.e., 36 x 2 x 3) stimuli were needed.

The fillers for the categories in the logical structures of the instructions were a set of 15 highly familiar, one-word (or in two cases, two-word) descriptors. Ten of the descriptors were semantically opposed, creating five pairs of complementary antonyms. The total set of 15 descriptors, with antonym pairs given first, was: "male-female"; "married-single"; "white-black"; "righthanded-lefthanded"; "over 15-under 26"; "employed"; "homeowner"; "student"; "parent"; and "veteran".

The rationale for choosing these descriptors was (1) to simplify categories as much as possible to minimize the processing of content in relation to the processing of logical structure, and (2) to enable subsets of ten descriptors to be drawn such that half were true and half false for any particular subject.

Since the "truth value" and "exit position" dimensions depended upon the individual subjects' category values, it was necessary to obtain these values prior to constructing the stimuli. Ten descriptors were selected from the set of 15 fillers such that half were true and half false when applied to the subject. This was accomplished by asking the subject his or her truth values for the five non-paired descriptors ("homeowner," "parent," "veteran," "student," "employed") as well as the values for one member of each of the 5 antonym pairs. Based on the values obtained for the first five descriptors, the appropriate term was selected from each of the five pairs, such that there were five true and five false descriptors for each subject. The selected descriptors were then randomly "inserted" in the stimulus set with the restrictions implied by the truth values and exit position dimensions: half true and half false instructions; and equal members of early, middle, and late exit positions. These restrictions were applied to all levels of the other dimensions as well (e.g., half of the four-category instructions were true, etc.). Of course, no category was repeated within a given instruction.

<u>Unpracticed Condition</u>. For the <u>first</u> set of stimuli (presented with no practice), a single logical structure was selected: the affirmative version of the four-category "and" string (Type I) used in the main condition: "A, B, C, and D." The three formats were crossed with the two truth values to complete the preliminary stimulus set. Two replications for each truth value-format combination yielded a total of 12 stimuli for this condition.

Exit position could not be systematically varied. The pattern of individual truth values for "true" structures was fixed by the logic of conjunctive strings: "TTTT," with the exit on the last category. For "false" structures, two patterns were used: "TFTT," with an exit after two categories, and "TTTF," with an exit after four categories. Thus each block consisted of one two-category exit and three four-category exits. The four combinations of exit position and truth value were presented in fixed rather than random order within a block. Table 3 shows the patterns for the four trials in each block.

Table 3. Fixed Exit Positions and Truth Patterns for the Unpracticed Condition by Trial Order and Block Order

Block order	Trial order	Exit pos.	Individual truth patterns	Overall truth values
Block 1	1	4	T T T T	T
	2	4	T T T T	T
	3	2	T F T T	F
	4	4	T T T F	F
Block 2	1	4	T T T T	T
	2	2	T F T T	F
	3	4	T T T F	F
	4	4	T T T T	T
Block 3	1	4	T T T F	F
	2	2	T F T T	F
	3	4	T T T T	T
	4	4	T T T T	T

PRESENTATION OF STIMULI

The experiment was conducted on a 2640B display terminal connected to an HP3000 computer system. This particular terminal imposed some technical constraints on the graphic presentation of the stimuli, primarily in terms of the possible layouts of the flowchart algorithms. However, the greater constraint on alternative presentations of the instructions was the need to optimize the layout of each format for ease of processing. Only in this way could we achieve a fair comparison of prose, flowcharts, and lists. Yet little is known about what constitutes optimal design in each format. That design appears to depend on a

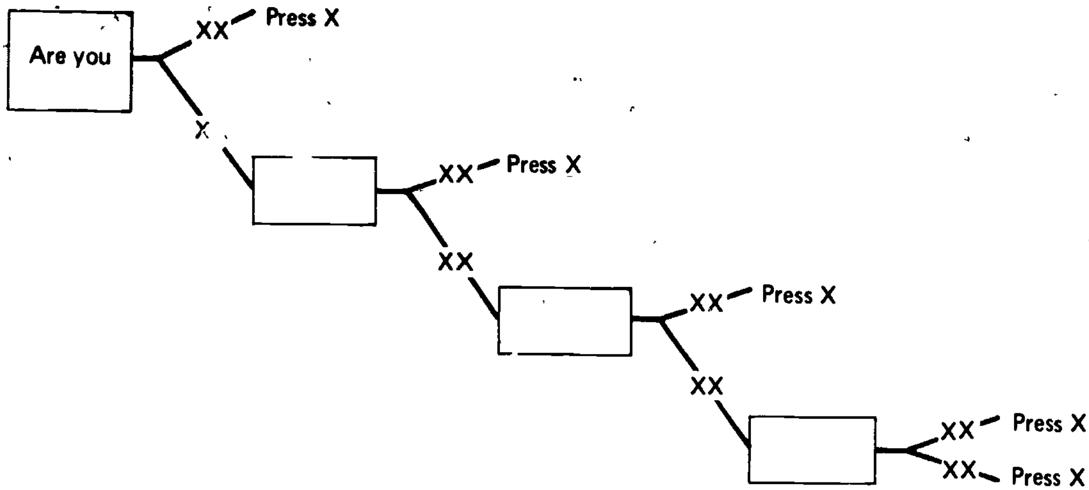
complex interaction of the task, the user, and the structure of the instruction. Our choices for layout were therefore intuitive, based on observing a pilot sample of alternative layouts in each format, and judging what appeared to us to be the clearest arrangement within the space and feature constraints of CRT presentation.

For prose the most obvious simplification was to trim words, clauses, and sentences to the shortest units possible.* Further, for graphic simplicity, we decided to display prose with the sentence centered in the screen and the first word located near the screen margin to the left of center. Longer sentences were continued on two lines (at most), with the second line beginning at the left margin immediately beneath the first line. Uppercase letters appeared at the beginning of the first word of each prose instruction, following normal sentence form. Underlying the different prose stimuli, there was one basic sentence template. Manipulating the number and type of connective words in this template generated the twelve structural frames required for this experiment, into which category fillers were inserted to make a complete instruction.

The flowchart design presented a wider range of choices. Many of these variables have never been specified; few have been tested for whether and how they affect human performance. Formal variables include multiple vs. binary branches (arrows), and question plus "yes-no" branches (as in Figure 1) vs. branches captioned with category choices. Visual-graphic variables include left-right vs. top-bottom flow direction, open vs. boxed (framed) questions, arrows vs. "rivers", and a number of other size, placement, orientation, and color choices.**

The display medium plays a more critical role with flowcharts than with prose in determining what options are possible on dimensions such as these. With little information available on the optimal design of flowcharts, we based our decisions on viewing several possible layouts on the computer screen. The final choices can be seen in the sample of templates shown in Figure 5. Though reduced in scale, these templates are identical in proportion, in distribution of upper and lower case letters, and in other features to the templates displayed on the CRT screen.

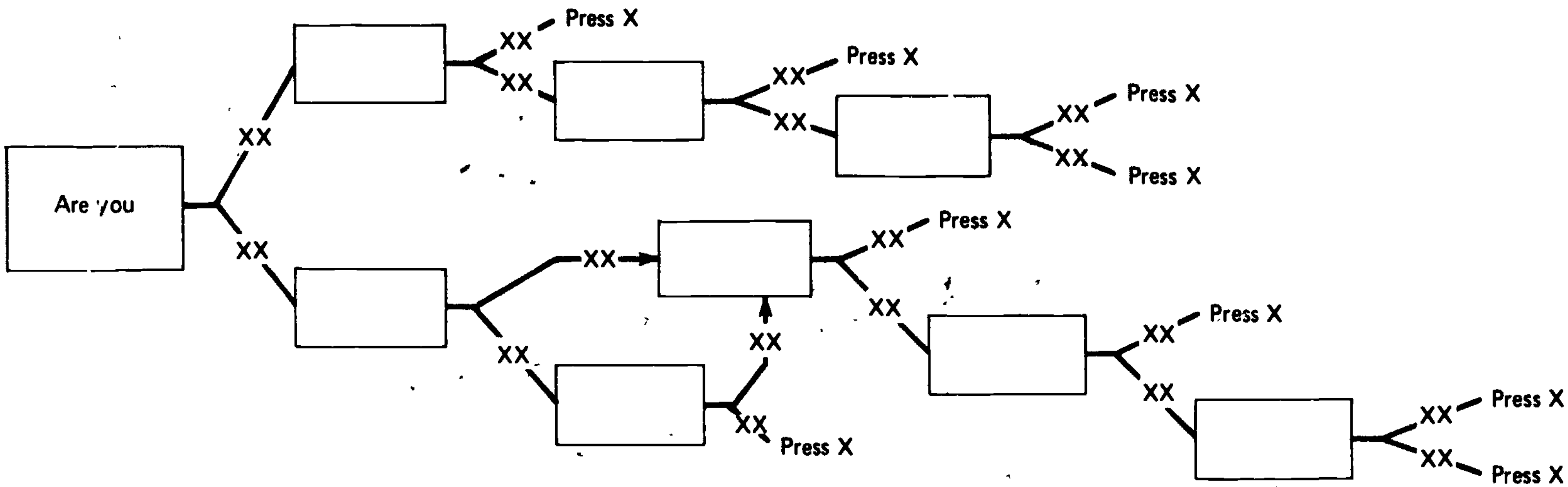
*Another possible simplification was to break the prose instruction into two or more short sentences. Without testing this option, we speculated it would do little to simplify the processing task: short sentences still require truth-functional connectives between them to capture the underlying truth table.
**Kammann (1975) described and tested some of these formal and graphic options.



a. Four-Category String and Four-Category "Not" Grouping

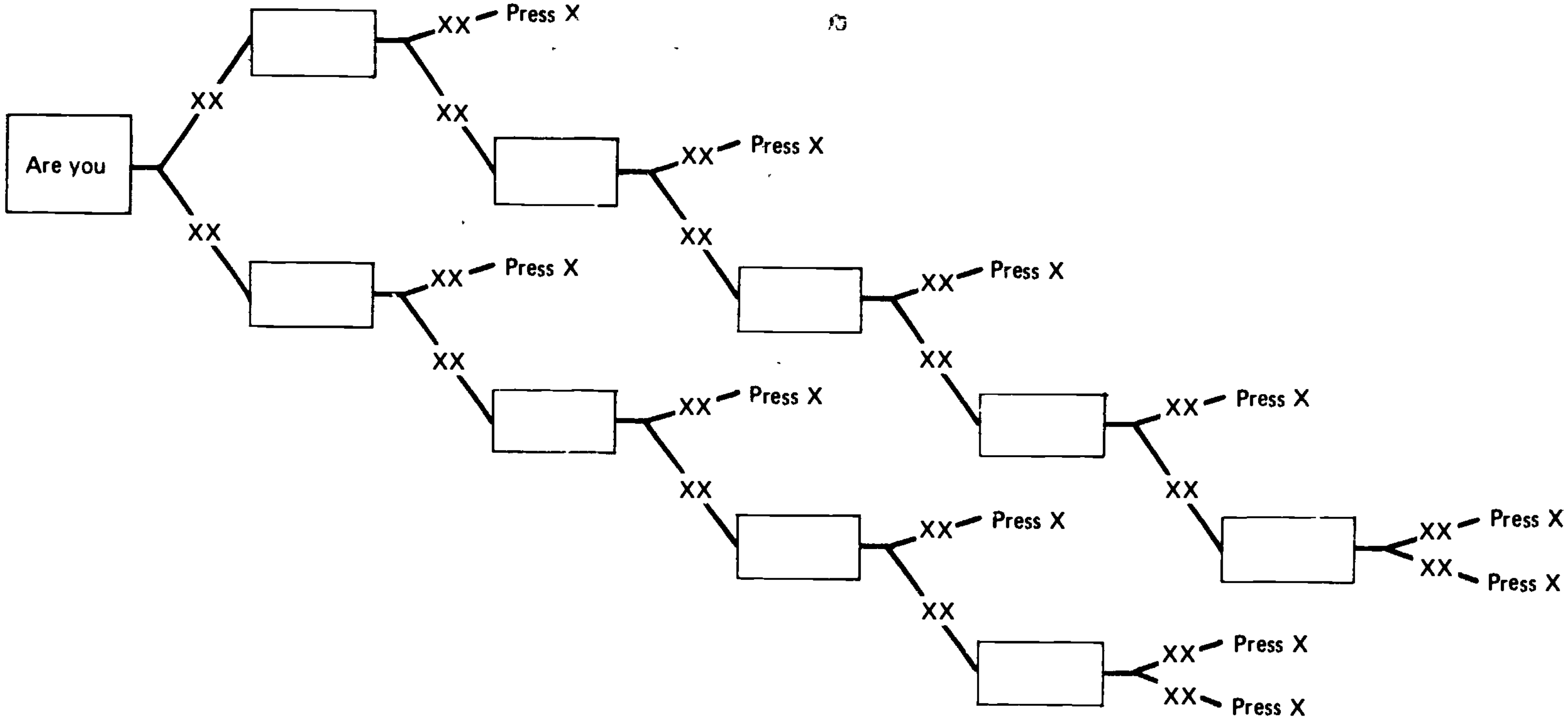
FIGURE 5. Selected Flowchart Templates. *

* The question boxes were reverse field, with dark letters on white grounds. A question mark ("?") was placed after each word or phrase inserted in a question box.



b. Six-Category Single Grouping

FIGURE 5. Selected Flowchart Templates. * (continued)



c. Six-Category "Not" Grouping

FIGURE 5. Selected Flowchart Templates. * (continued)

Six templates were required in the flowchart format to represent the 12 structures of the stimulus set.* Figure 5 shows three of these templates. Organizational complexity can be seen to affect directly the complexity of the template's shape. In general, flowcharts for strings are linear; those for groupings are branched. Note, however, that the flowchart for four-category "not" groupings duplicates the linear flowchart for four-category strings. This is because the "not A, not B, and not either C or D" structure is the logical equivalent of "not A, not B, not C, and not D." The latter, linear organization is the more basic and determines the shape of the flowchart.

Flowchart templates were constant for variations in logical structure other than organization--i.e., "and" vs. "or" and negative vs. affirmative. The flowchart reflects the two latter dimensions in the assignment of labels to arrows or commands rather than in spatial characteristics. The labels that marked arrows were either "yes" or "no" (indicated as "xx" in Figure 5). The labels marking command outcomes were "press *" or "press $" (indicated as "press x" in Figure 5). The "and-or" distinction was mapped onto the flowchart by both (1) interchanging the "yes-no" labels on the arrows; and (2) interchanging the labeled outcomes "Press $" and "Press *."

Lists offered fewer formal and graphic choices than flowcharts--these choices were largely in how to space lines and items. Again, for those variables that seemed significant to processing, we made decisions intuitively by observing the choices on the CRT screen.

There were two templates for the "jump" list format, depending on the number of categories: a list of four "Are you ___?" questions for four-category structures; a list of six questions for six-category structures. The four-category template is shown in reduced form in Figure 6. The proportions and other features are identical to those of the template that appeared on the CRT screen.

*Each flowchart template was a set of boxes and arrows distinguished by shape--that is, by the number and the configuration of boxes. Defined on these criteria, templates varied with (1) the number of categories in the conditional clause and (2) the organization of the categories--string vs. grouping, and number of branches in the grouping.

1. Are you . . . . If no, go to X.
If yes, go to X.

2. Are you . . . . If no, go to X.
If yes, go to X.

3. Are you . . . . If yes, go to X.
If no, go to X.

4. Are you . . . . If yes, go to X.
If no, go to X.

5. Press X

6. Press X

FIGURE 6. List Template for Four Categories.*

* A question mark ("?") appeared at the end of each word or phrase inserted in the blanks in a template. The spacing between the end of the inserted word or phrase and the beginning of the "If" command was always three characters.

Although organizational type does not have the overt effect on spatial structure that it did with the flowchart, it affects the "jump patterns" involved in following "go to" commands. Jump pattern refers to the number and the size of skips between items in a route through the list. These two variables are determined by the combination of organization, length, and exit position.

The 12 frames necessary for the list condition were generated from the two list templates by inserting and manipulating the "yes" and "no" responses, the item numbers specified in "go to" commands, and the key specifications in "Press" commands.

Finally, for all three types of formats, most graphic characters appeared in white on the dark field of the CRT screen. However, the boxes of the flowchart were presented in reverse-field: the area of each box was white while the words within boxes were in dark letters.

III. METHOD

Subjects. Fifty-four adults, 20 males and 34 females, served as subjects. All were fluent English speakers. Half were undergraduate and graduate students from three universities in the Washington, D.C. area, and were paid $5.00 for their participation. The rest were employees of the American Institutes for Research, who donated their time. The employees were from both clerical and research positions, and all had completed at least a bachelor's degree.

Experimental Design. To recapitulate, the experiment used a mixed within-between factorial design to assess response time and accuracy. The within-subject factors were: (1) format (3 levels), (2) number of categories (2 levels), (3) type of connective (2 levels), (4) truth value (2 levels), and (5) exit position (3 levels). These were combined factorially to produce 3 x 2 x 2 x 2 x 3 = 72 instructions. There were no replications. The between-subject variables were (1) organizational type (3 levels) and (2) order of format blocks (3 levels). With respect to the latter, subjects were randomly assigned to one of three orders, determined by the identity of the first block: prose first, flowchart first, or list first.

The unpracticed condition used a 3 x 3 x 2 factorial design, retaining format and truth value as the within-subjects factors and order of format blocks as the between-subjects factor. (Exit position was treated as a random variable.)

Finally, the content of categories (fillers) and the order of presenting instructions within a block were treated as random variables, in both the unpracticed and main conditions.

Implementation. A computer program generated the conditional instructions in an interactive session with each subject. The program was designed to:

- assign subjects randomly to the nine between-subjects cells created by the combination of organizational type and order of format blocks;
- select the 10 words representing half true and half false descriptors for each subject;
- place fillers in the instructional frames according to the stimulus generation procedures described above;

- order the presentation of stimuli randomly within format blocks, subject to the restriction that instructions of the same length or same organizational type could occur no more than three times in succession;

- record and score response time and accuracy.

The selection and sequencing of fillers in frames was randomized independently between trials (the presentation of an instruction), and between subjects. The order of presenting the instructions was randomized independently between subjects and between format blocks.

Procedure. Subjects were tested individually in a single session lasting 50-60 minutes. Each subject responded to 72 instructions, all of a single organizational type, presented one at a time on the screen of the HP display terminal. Subjects sat at the keyboard of the terminal, where they indicated their responses by pressing one of two keys, labelled "*" and "$". These keys were about 1.5 inches apart. The rest of the keyboard was covered. Subjects were instructed to respond with the index finger of each hand throughout the trials. The subject's hands rested on the keyboard panel between trials, while subject's two response fingers were poised above the response keys.

Subjects were read introductory instructions explaining that "this was a study to see what is the best way to present instructions to make them easier to follow." They were asked to read and respond to each instruction by pressing the appropriately labelled key--either "*" or "$". Subjects were told that sometimes they would see a sentence that said only what key to press if the condition were true for them. (This was the case for all prose formats.) If the condition were false, subjects were to press the remaining key. Two short, one-category sentences were presented to illustrate this procedure.

Subjects were told they would see instructions "in different forms, some of which might be unfamiliar," but they were not told what forms they would see, nor given any directions on how to handle the different formats. They were told that some instructions might be hard, but to do their best--that the point was to try to read and answer as quickly as possible while still being as accurate as possible.

Before the trials began, subjects saw 10 questions on the screen representing the 15 categories ("Are you male?" etc.). They were asked to answer "yes" or "no" while the experimenter typed in the answers. The program then selected 10 descriptors (out of the 15 possible) yielding an equal number of "yesses" and "noes" and presented these 10 to the subject while the experimenter explained, "These are the words you will see in the

sentences to come." The subject was asked to confirm that the designation of these descriptors as true or false was correct before the experimenter proceeded.

The subject was then presented the 12 "cold" trials in the unpracticed condition. The instructions in these trials were presented in sets of four, blocked by format.

After these 12 trials, subjects were given a chance to ask questions. The experimenter answered questions about response procedure and purpose, but not specifically about how to interpret the prose logic or how to follow the algorithms.

Subjects were then told they would see instructions like those they had just responded to but somewhat more complicated. They were informed that these instructions would be presented in groups based on the form of the instruction, as with the groups they had just seen; but that there would be many more instructions in each group d that each group would be preceded by eight instructions f practice.

Subjects were then presented the three blocks of 24 experimental trials each. Eight practice trials (with feedback) in the appropriate format prefaced each block, producing 32 trials in each format. The program signalled "end of practice" between the practice and the experimental trials. Subjects could ask questions at this point, but the experimenter's answers stayed within the limits set for the unpracticed condition.

The practice instructions sampled each of the four logical structures (four vs. six categories, "and" vs. "or") included in the organizational type assigned to that subject. Structure was counterbalanced with truth value to yield the eight practice frames. The placement of fillers in these frames was randomized by the same procedure used in the experimental trials.

Subjects received feedback on the practice trials in the form of a programmed "incorrect" signal that flashed on the screen immediately after an incorrect response. The misperformed instruction then reappeared. This procedure served to stress the accuracy constraint.

Each trial began with the signal "READY" flashed on the center of the display screen for about two seconds. From one to two seconds after the offset of the signal, a conditional instruction appeared and remained on the screen until the subject pressed a response key, terminating the trial. If the subject had not responded after 60 seconds, the sentence disappeared and the subject was forced to select a response.*

*There were no cases of latencies this long during the experimental trials.

A variable interval of 5 to 16 seconds separated the trials. The variable interval was necessitated by the software requirement of writing each instruction on-line before it appeared on the screen. The writing procedure took 10 seconds for the four-category flowcharts, 16 seconds for the six-category flowcharts, 5 seconds and 8 seconds for the four- and six-category lists (respectively). These durations determined the length of the respective intertrial intervals for the flowchart and list formats. For prose, the writing duration was much shorter. Therefore, a 5 second interval was imposed to make these trials more consistent with those of the other two formats. Subjects were told simply to expect a variable wait, which would sometimes be rather long, between trials.

A rest period of approximately 30 seconds separated the three trial blocks, longer if the subject desired. The experimenter remained in the room with the subject throughout the session.

Subjects were timed in hundredths of a second from the onset of an instruction to the first push of a response button. Response accuracy was scored by an algorithm which verified each antecedent statement according to its coordinate structure in conjunction with a subject's truth values on the individual words.

At the end of the session, subjects were asked to rank the three formats according to which they liked the best. Subjects were also asked background information: level of education, professional or academic field, and experience with a logic and/or programming course.

IV. RESULTS

1. Main Condition

We analyzed the data from the main condition ( ) test trials) in two steps. First, we conducted ANOVAS on the e .ire 3 x 3 x 2 x 2 data set (organization x format ~ ' th x and-or), collapsing over truth value and exit positio. ..e ANOVA was performed on mean response time, one on mean number correct.** Both analyses showed significant three-way and four-way interactions. The major implication of an interactive pattern of results is that conclusions of the sort "format X is better than format Y" cannot be drawn; instead, statements about format differences must be anchored to the logical structure of the instruction.

The second step, then, was an attempt to untangle the complex interactions by conducting separate ANOVAs for each of the three organizational types--strings, simple groupings, and "not" groupings--since organizational type was a between-subjects factor in this design. Thus six new ANOVAs were conducted, each a 3 x 2 x 2 design (format x length x and-or), three each for response time and for number correct. These ANOVAs are shown in Appendix A.***

It is clear that the pattern of results was still interactive. Rather than overload this presentation with a discussion of each significant interaction, we will present those data that yield the clearest answers to the first three motivating questions of this study: Are algorithms better than prose? What are the limits on this conclusion? How do lists and flowcharts compare? We will link the relevant data to a model of how people process complex conditional instructions in each format.

The first question posed in this study was: For instructions with truth-functional conditions, do algorithms improve performance over prose? The clearest answer to this question is provided by the graphs showing lists, flowcharts, and prose for each of the three organizational types, collapsing over the remaining stimulus dimensions. Figures 7a-c present these plots for response time; Figures 8a-c present the plots for number correct. There are separate graphs for strings, simple groupings,

---

*Order of format blocks was controlled by counterbalancing in the design.

**Two ANOVAs for response times were conducted using both raw data and logarithmic transformations to normalize the positively skewed distributions of time scores.

***Since clearer interpretations were possible off of the transformed data, the appendix shows only the ANOVAs for those data. The transformed and untransformed data yielded similar patterns.

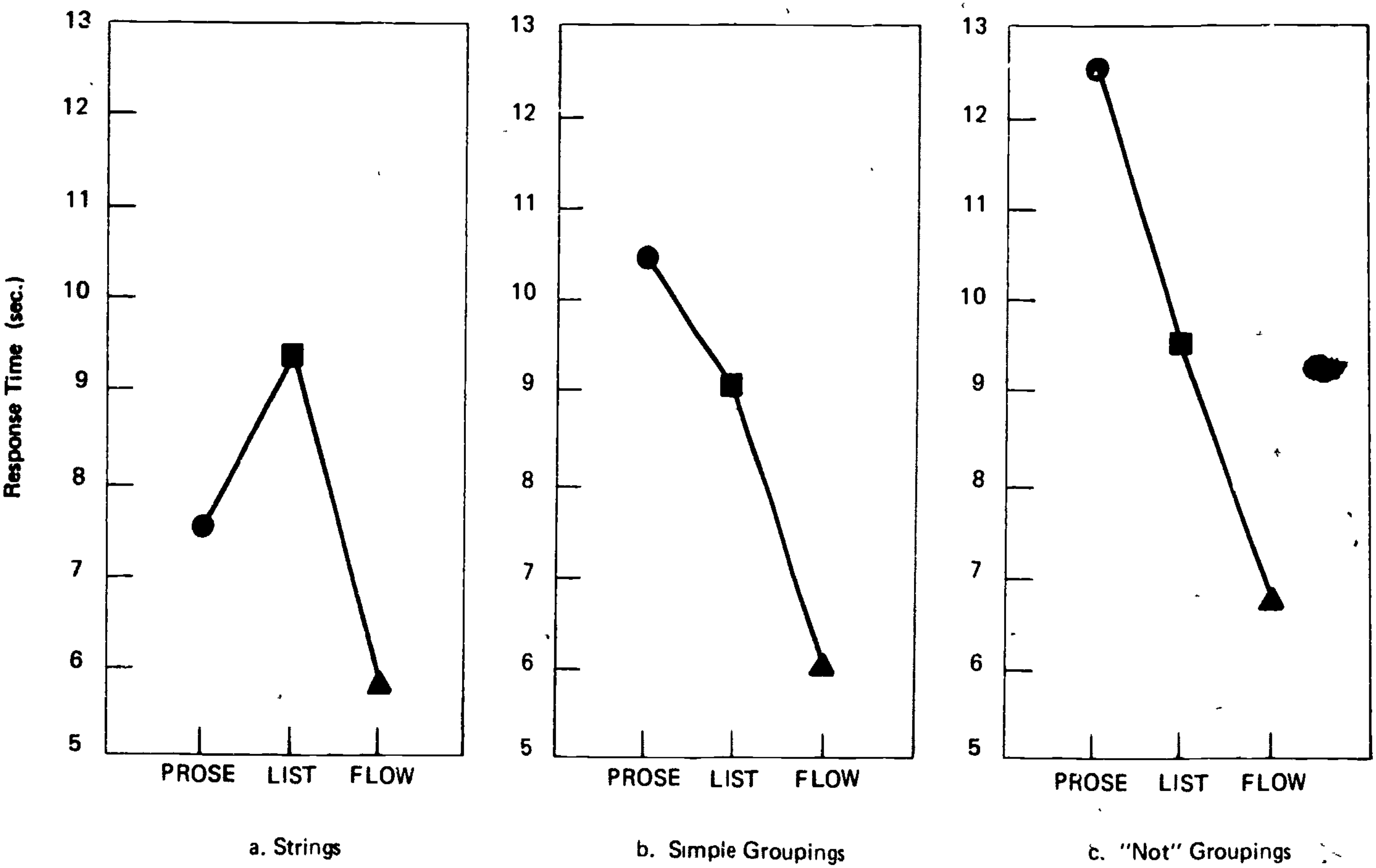


FIGURE 7. Mean response time as a function of format for each organizational type.

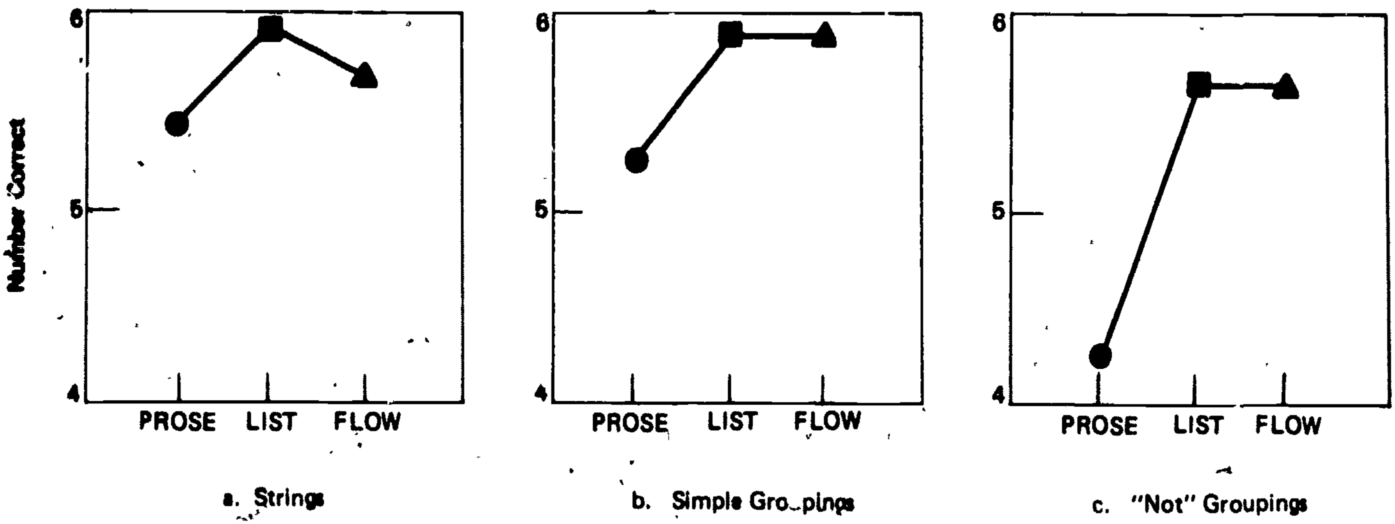


FIGURE 8. Mean number correct as a function of format for each organizational type.

and "not" groupings. These graphs show that processing of lists and flowcharts is generally both faster and more accurate than processing of prose.

The second question posed in this study--where and how much do algorithms improve performance over prose--requires a finer-grained analysis. From the graphs, we can see that there are exceptions to the superiority of algorithms, as well as variations in degree, depending on a combination of factors: the organizational type, the measure, and the form of the algorithm. For example, when the structure is a string, list algorithms have slower response times than prose. Furthermore, we can infer that prose is sensitive to logico-structural changes in the stimulus instructions while algorithms are not. Prose becomes slower and more error-prone as the organization goes from easiest (strings) to hardest ("not" groupings), while algorithms maintain relatively uniform speed and accuracy across organizations. Indeed, algorithms appear uniformly nearly error-free.

To understand these variations in the relations among formats, we must look at what underlies performance. Our best explanation of performance is in terms of an informal theoretical model of what readers do in each format. This model also provides the grounds for answering our third question--how do lists and flowcharts compare with each other?

We describe this model below. Our presentation omits several assumptions necessary to qualify the model as formal: it is intended more as an organizing heuristic for the complex experimental results than as a tool to generate precise quantitative predictions.

Algorithms

The model. Our best explanation of performance with algorithms is a linear processing model that predicts response time in terms of the exit position in the stimulus instruction: that is, the minimum number of categories that must be processed to verify the condition and determine a response to the instruction. Exit position is the dimension hypothesized to be uniquely pertinent to performance with the algorithms, based on the obligatory partial processing that characterizes that format. We can represent the effects of this dimension with three parameters, signifying three basic kinds of operations required to process an algorithmic instruction. We assume that each operation takes a fixed amount of time, identified by that parameter.

42

The first parameter, which we will call "r," represents the reading time for a given step in the algorithm.* This is the time needed to encode and answer a given question, and to match the answer to the appropriate "yes" or "no" symbol (which indicates the way to the next question or outcome in the sequence). The second parameter, which we will refer to as "m," represents the moving time for a given step in the algorithm. This is the time needed to make the physical transition between one question and another or between a question and final response (outcome). The third parameter, "e," represents the time needed to select and execute the final response to the instruction.

By these definitions, the values of "r," "m," and "e" should change with the type of algorithm, since all parameters are determined by the nature of the physical and symbolic connections between steps. We will postpone consideration of these parametric variations until we discuss the flowchart-list comparison.

To complete the model, we assume that subjects process only the questions (categories) presented in that route through the algorithm that is relevant to their particular combination of truth values. Assuming the relations of a linear processing model, it follows that the operations of reading and moving occur once, additively and independently, for each category represented in a given route. The execution parameter "e" occurs only once per instruction. To produce the final response time for an instruction, the parameters "r" and "m" would be multiplied by the number of categories in the route and added to "e." Since the total number of categories in a route is identified by the exit position variable, this model predicts a constant, linear rise in response time with increasing exit position. Response time should not vary over changes in the logical structure and the length of the instruction.

The response time data. The data germane to this model are the breakdowns by exit position of the response time results. Appendix B displays these breakdowns, showing response time for each combination of format x organization x and-or x length x exit in the data set. Since exit possibilities depend on truth value as well as length and logical structure, the truth value breakdowns are also given in Appendix B.

To test the model against these data, we performed regression analyses on the response time data in each format.** That is, we

*Let us consider a "step" the operation of going from the start of one question in the algorithm to the start of the next relevant question.

**As mentioned previously, it was impossible to include "exit position" as an experimental factor (i.e., factorially cross it with all other variables). Thus, the "tests" discussed here are all post-hoc.

calculated the best-fit regression equations for the exit position by response time functions and tested for the significance of the linear component. Results for the overall analyses appear in Table 4. It is clear that exit position is a potent predictor of response time in algorithms: The overall correlation between response time and exit position was r = .937 for lists, r = .796 for flowcharts. This means that response time in each format increased as a relatively uniform, linear function of exit position, irrelevant of length or structure. This relationship conforms to the predictions of the model. By contrast, the correlation between exit position and response time for prose was r = .006, demonstrating the model's assumption that the partial processing effect is unique to algorithms.

TABLE 4.
Response Times for Exit Positions in Lists and Flowcharts

Exit Pos. [a]	(n) [b]	List	Flowchart
1	(7)	5.03 (.73) [c]	3.79 (.30) [c]
2	(10)	7.39 (1.05)	5.11 (.98)
3	(9)	8.35 (.65)	5.57 (.73)
4	(32)	10.23 (.81)	6.77 (1.21)
6	(14)	14.29 (1.51)	9.10 (1.73)
Mean		9.89 (2.88)	6.55 (1.96)
Correlation		.94	.80
Slope		1.79	1.04
Intercept		3.27	2.72

NOTES:

a mean exit position = 3.694

b number of instructions per format with given exit position

c standard deviations in parentheses

To clarify these analyses, we have plotted the estimated regression lines for predicting response time from exit position. The regression lines for lists and flowcharts are displayed in Figures 9a and b (respectively). These lines show how each additional relevant category in the instruction adds a constant processing load, given by the slope of the line. The lines

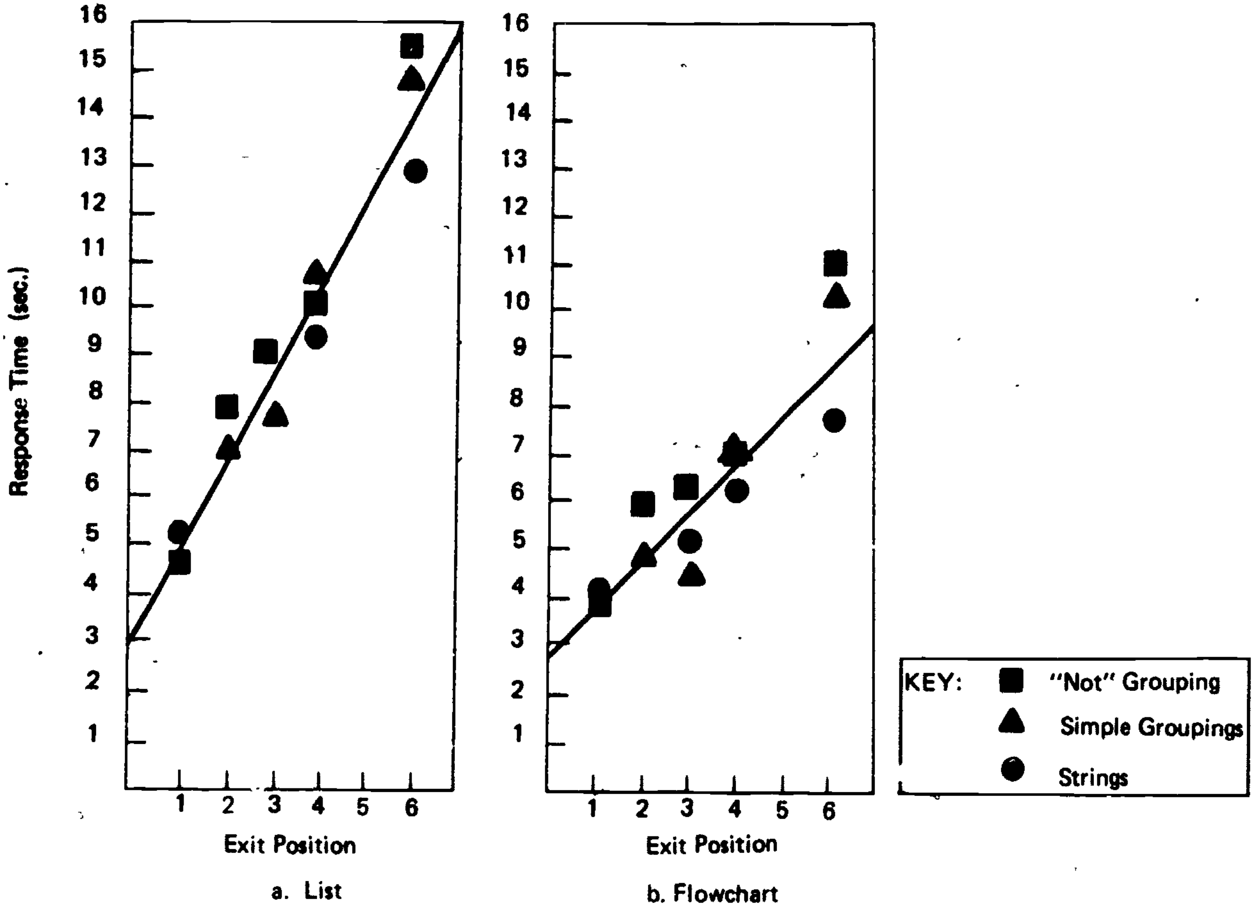


FIGURE 9. **Estimated regression lines for predicting response time from exit position for lists and flowcharts. (Mean response times for organizational types are plotted around each line.)**

further clarify the list-flowchart differences in slope and intercept indicated in Table 4. We will return to these differences in considering the question of lists vs. flowcharts.

The data points plotted around the regression lines in Figure 9 show the obtained mean response times for each of the exit positions broken down by organizational type. These points cluster closely around the two lines, reflecting the strong linear association between time and exits overall and supporting the model's prediction that logical variables will not affect performance and algorithms. Nevertheless, there is a certain amount of response variability attributable to logical organization, and we will look at the breakdown of the time-exit correlations by organizational type at a later point in this presentation.

The number correct data. The accuracy results presented in Figure 8a-c are self-explanatory. There appear to be no significant errors with algorithms. Supported by the response time data, these results demonstrate that algorithms virtually eliminate the logical asymmetries prevalent in performance with prose. There was a slight drop in number correct for "not" groupings in both lists and flowcharts (Figure 8c). Separating the number correct data by four vs. six categories showed that this drop is attributable to the six-category, branching template. We will return to this issue later in this presentation.

The absence of errors allows several inferences about how people performed with algorithms in this experiment. Earlier we hypothesized that category-level operations (reading and answering individual categories) would not create errors since the categories are simple and overlearned. The data support this hypothesis and also allow us to infer that the other operations involved in the algorithm task--encoding, matching yes-no answers, following arrows, scanning, searching and responding--are negligible sources of error. Although these operations can be time-consuming, as in the list, they are nevertheless simple and explicit and unlikely to produce mistakes.

Prose

The model. The results for prose are more complex than those for algorithms. Figures 7 and 8 demonstrated the unique sensitivity of prose to organizational structure. A further breakdown of the prose data by "and-or" and by four vs. six categories, Figures 10 and 11, reveals the unique influence of logical connectives and length. These results are consistent with our hypotheses about the effects of truth-functional variables on performance with prose.

40

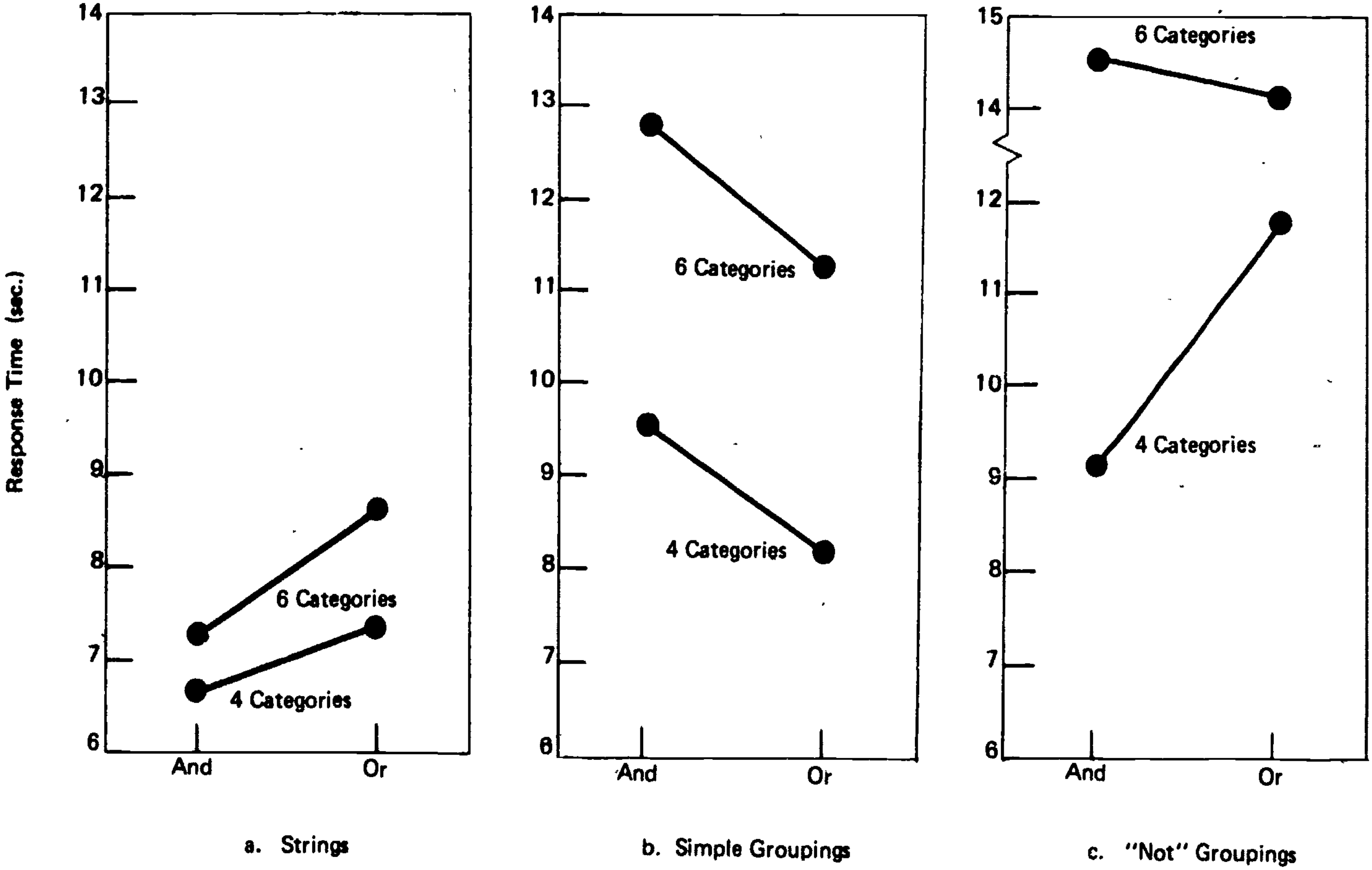


FIGURE 10. Mean response time for prose as a function of "and-or" for each organizational type.

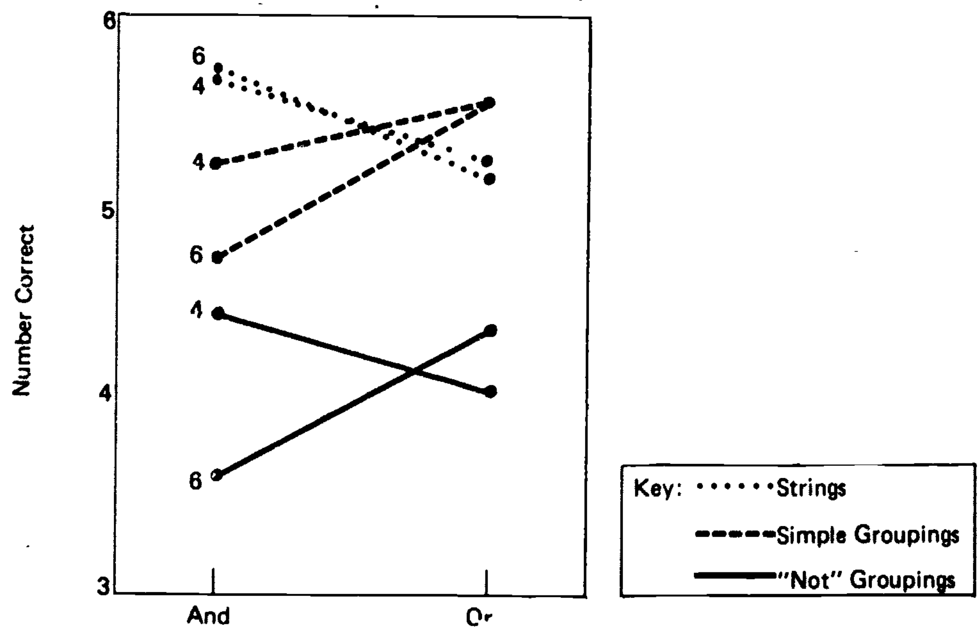


FIGURE 11. Mean number correct for prose as a function of "and-or" for each organizational type.

Our best explanation for the interactive relationships obtained in the ANOVA and shown in these figures is in terms of a processing model built on parameters of logic and length rather than exit position. This model is based on the task analysis for prose instructions described earlier, and serves both to organize the hypotheses drawn from that analysis and to give a general account of the kinds of logical variables that make prose more or less difficult and more or less subject to improvement by algorithms.*

We will sketch the model with respect to response time. Extending the linear relationships assumed by the algorithms model, the prose model conceives the instructional task as a series of cognitive operations combining additively to produce the final response time. This model makes a basic processing assumption: subjects exhaustively process the prose instruction, reading all categories presented in the antecedent clause.

The "e," "m," and "r" parameters retain their conceptual meaning, although they presumably have different values for prose. The other components of response time in this model derive from logical operations, unique to prose. In our original processing model (Holland and Rose, 1980), we captured the effects of truth-functional complexity on performance with three parameters, representing the additional mental operations required (1) for "or" connectives over "and," (2) for comma-implied (e.g., "A, B, or C") over explicit "or" (e.g., "A or B or C"), and (3) for grouping over linear structures. Here, we add a fourth parameter, representing the operation required to deal with negation.

These predictions follow: Response time should be unaffected by the exit position of an instruction but should increase as a function of increasing length and the additive combination of any of the four logic parameters just described. For predictions about accuracy, we assume that category-level errors do not occur in performance with these instructions. but should instead be a function of the logical parameters. The model thus predicts that errors in prose will be unaffected by length per se; rather, errors will increase with increased logical processing demands. Thus, error effects will be identical to those in response time, disregarding length.

Let us compare these predictions with the observed response time and errors.

<u>The response time data</u>. The first observation concerning response time, from the regression analyses, is the virtual lack of correlation between response time and exit

*A full description of this model is given in our previous report (Holland and Rose, 1980).

position for prose (r = .006). On the other hand, the correlation between response time and length is moderate, at r = .433. These observations support the assumption of exhaustive processing posited in the performance model.

Returning to the functions relating format to organization (Figure 8), we find that response time for prose increases over organizational types, with strings (Figure 8a) taking the least amount of time and "not" groupings (Figure 8c) taking the longest. This ordering is explained by the addition of the grouping parameter between strings and simple groupings, and the addition of the negation parameter between simple and "not" groupings.

Looking within organizational types, we find further variations in prose response time as a function of the number of categories and the pattern of connectives. These variations can be seen in Figures 10a-c, which show the prose functions relating length to the "and-or" variable for each organizational type.

The first salient feature in these graphs is that six-category structures are consistently slower than four-category structures at each organizational level. This delay reflects the addition of the reading parameter ("r") with added categories. The second notable feature is the "and-or" asymmetry displayed at each organizational level. These asymmetries reflect the constant effect of the "or" parameter: In general, the presence of "or" in an instruction increases response time incrementally for each group in which the "or" occurs.

There are two apparent anomalies in the shape of the "and"-"or" and four vs. six functions that we will address briefly. First, the slope of the "and-or" function is positive for strings, negative for simple groupings. This reversal is explained by how we have defined "and" and "or" structures. For groupings, "and" structures are those with the major groups connected conjunctively: "A or B, <u>and</u> C or D." Thus, "ors" outnumber "ands" in "and" groupings.* The second peculiarity of the "and-or" functions in Figure 10 is the failure of the six-category "not" grouping to show the predicted "and-or" asymmetry (Figure 10c). The predicted asymmetry <u>did</u>, however, emerge strongly in response errors, as we will see in examining the accuracy data. The simplest explanation is that subjects were sacrificing accuracy for speed in this, the most difficult condition. Subjects may have imposed a ceiling of around 14 and a half seconds on response latencies.

*In the third organizational type, the six-category instruction is a grouping, with the "and" structure predicted hardest, while the four-category instruction is more linear, with the "or" structure predicted hardest.

50

There are further complexities in the graphs in Figure 10. For example, the slope of the "and-or" function in both types of groupings (except for the six-category "not" grouping) are steeper than the slope of that function in strings. Similarly, the difference between four and six categories in both types of groupings is far greater than that difference in strings. We will not attempt to explain these variations fully here; suffice it to say that they are in large part predicted by the logical processing parameters.

However, there are some additional effects not attributable solely to truth-functional complexity. For example, in the third organizational type, the difference between "and" and "or" at four categories is nearly three seconds--greater than any other "and-or" difference in the stimulus set. What underlies the unprecedented difficulty of the "or" instruction here? We will offer a speculative account.

The instructions in question have these schematic structures: for or--"not A, not B, or not both C and D;" for and---"not A, not B, and not either C or D." The former structure incorporates two of the parameters hypothesized to complicate processing: the "or" operation as the major connective, and the presence of an implied "or" in a comma series.* Furthermore, the four-category "not" grouping is the only structure in the entire set in which all categories are negated. It may be that the difficulty of "or" series with commas is exacerbated by (or interacts with) negation.

Thus an explanation for the difficulty of the "not A, not B, or not both C and D" instructions may start with a combination of these factors: the difficulty of the implicit "or" series, and the enhancement of this difficulty by "nots."

In conclusion, the shape of the prose data can be understood largely in terms of the processing model we have sketched, which predicts the difficulty of prose instructions in terms of a linear combination of logic and length parameters. Thus the fastest response in the stimulus set occurs with four-category "and" strings, as shown in Figure 10, because these stimuli contain neither the logic parameters, nor the extra category reading operations, hypothesized to add increments of response time.

*The explanation posited in the processing model in Holland and Rose (1980) was that subjects represent implicit series (A, B, C, or D) as if the commas were "ands." This appears to be a fundamental encoding bias. Subjects must then transform the sentence or group into a disjunction when the "or" is discovered. This transformation time is over and above the original increment required to verify disjunctive as opposed to conjunctive structures.

The number correct data. Looking briefly at the accuracy results in Figures 11a-c, we find that the number correct functions across organizations and connectives are as predicted in the model: The "and" string is virtually errorless; accuracy declines with added components of logical difficulty.

With respect to logical factors, then, the accuracy relationships correspond at most points to the time relationships. The "and-or" asymmetries in errors directly reflect the asymmetries in time, as seen in comparing Figure 11 with Figure 10. It appears that subjects in this study handled logical difficulty by both slowing down and committing errors.

Figure 11 also suggests that length per se does not affect accuracv, further supporting the modeling assumption of no category-level mistakes. This suggestion is particularly clear for the strings in Figure 11. The exceptions are those structures where added categories entail new components of logical difficulty. For example, the simple "and" grouping accrued more errors at six than at four categories. Briefly, the distinction between the four-category structure (A or B, and C or D) and the six-category structure (A, B, or C; and D, E, or F) is the reorganization of binary "or" branches (A or B) as trinary "or" branches (A, B, or C). As posited in the model, the trinary branches are more difficult (beyond the fact that they contain more categories) because they contain an initial comma expressing "or" implicitly.

Summary: Prose vs. Algorithms

Time. Comparing the prose model with that for algorithms shows why the advantage of algorithms for response time varies with the logical structure and length of the instruction, as well as with exit position. The model for algorithms posits that the processing of algorithms stays constant with changes in logical structure.* It follows that the harder the structure of the prose instruction, the more algorithms should improve performance. Also, exit position affects the processing time of algorithms but not of prose. It follows that the advantage of algorithms shall diminish with later exit positions.

*Note that algorithms do reflect logical structure in terms of the minimum number of categories involved in different partial processing routes through the algorithm. The exit possibilities in an instruction are an inherent function of structure and truth value. For example, in our description of the stimulus set, we showed how the minimum exit is earlier for strings (exit = 1) than for hierarchies (exit = 2 for 2-branch hierarchies).

The models do not predict which of these effects (logical variables or exit position) is "stronger"--or, more simply put, when algorithms become superior. In general, for most of the instructions examined in this study, algorithms were faster than prose regardless of exit. An interesting comparison, however, arises in the relatively easy "and" strings. If the exit position was one, then the flowchart was a little over three seconds faster than prose. If the exit was four, then the two formats were equally fast. Moreover, the list was about two and a half seconds faster than the prose "and" string at the early exit, but about two and a half seconds slower at the late exit. The obvious implication is that there are some conditions where prose instructions are no worse than flowcharts, and are preferable to list algorithms. In terms of our models, these conditions occur when "encoding and moving" time take longer for the algorithms and logical difficulties are negligible. We will explore this general issue as part of the discussion of the "unpracticed" condition in a later section of this report.

Errors. Comparing the accuracy data for prose and algorithms, we can see that all instructions except "and" strings are improved by algorithmic translation. Again, the processing model which predicts virtually errorless performance with algorithms, and the model which predicts prose errors as a function of logical parameters, account substantially for the accuracy data in this study.

Together, the models we have described explain the significant interactions in the ANOVAs on response time on number correct, and the relative ordering of prose and algorithms observed in the plotted data.

Lists vs. Flowcharts

The third question addressed in this study--how flowcharts and lists compare--has to some extent been answered in the preceding presentation. It is clear from the graphs in Figures 7 and 8 that lists are slower than flowcharts, while both are equally accurate. We will attempt to formalize and explain this relationship in terms of the processing model for algorithmic instructions.

In our original characterization of this model, we assumed that the values of both the "e," "m," and "r" parameters were format-specific. How do these values vary between lists and flowcharts?

The reading parameter "r" represents a cluster of operations which in the flowchart includes reading the question, deciding the response, reading the "yes" and "no" answers, and matching the internal response with the appropriate written answer. The list parameter "r" includes not only these operations but also the

operation of reading the "go to X" command, preceding the transition between items. Thus the value of "r" for lists should exceed that of "r" for flowcharts.*

The moving parameter "m" in the flowchart represents the time required to follow the arrow from the "yes" or "no" signal to the next question or outcome. In the list, "m" represents the time required to scan the vertical list of questions in search of an item number matching the number specified in the "go to" command. The specified number must be retained in memory during the search. Clearly, given the extent of both the memory operations and the physical eye movements required, the value of "m" should be greater for lists than for flowcharts. Finally, the "e" parameter, since it requires discriminating a separate "line" for the lists, should be greater there than for flowcharts.

To see how these parametric differences are reflected in response time, let us return to the regression analyses presented in Table 4 and Figures 9a and b. The regression lines plotted in Figure 9 clarify the differences in slope and intercept obtained between lists and flowcharts in these analyses. With estimated slope parameters of 1.79 seconds and 1.04 seconds for lists and flowcharts respectively, lists take longer to process by about 3/4 seconds for each relevant category in the instruction (given by the exit position). The estimated intercept parameters vary in the same direction as slope: 3.27 seconds and 2.75 seconds for lists and flowcharts respectively. This difference is attributable to the "e" parameter.

In our first presentation of the regression analyses, we noted that the predictiveness of the overall correlations could be improved when broken down by organizational type. These breakdowns are shown in Table 5. For lists the new correlations are .970, .951, and .949 for strings, groupings, and "not" groupings respectively (compared with .937 overall). For flowcharts the improvement is more dramatic: .960, .879. and .838 respectively (compared with .796 overall). These breakdowns expose several systematic variations in the predicted linear relationships which have important processing implications for lists and flowcharts.

An explanation for the variations from linearity is that the flowchart uniquely reflects changes in logical organization by changing shape: strings take linear flowcharts; hierarchies take branching. This distinction was seen in Figure 5. The branching flowchart is a potentially complicating factor in performance because it entails inconsistent reading direction. This effect

*In addition, in the particular templates used in this study, the list repeats the "Are you . . ." before each question, further inflating the value of "r." The flowchart states "Are you . . . " once.

TABLE 5.

Response Times for Exit Positions by Organizational Type for Lists and Flowcharts.

	Strings			Single Groupings			"Not" Groupings		
Exit Pos.	(n)[a]	List	Flow	(n)[a]	List	Flow	(n)[a]	List	Flow
1	(4)	5.20 (.97)[b]	3.80 (.28)[b]		–	–	(3)	4 [illegible] .28)[b]	3.78 (.39)[b]
2		–	–	(8)	7.20 (1.06)[b]	4.86 (.91)[b]	(2)	8.[illegible]8 ([illegible]70)	6.10 (.[illegible]1)
3	(2)	8.23 (1.1[illegible])	5.51 (.03)	([illegible])	8.13 (.29)	5.16 (.69)	(3)	8.75 (.77)	6.15 (.76)
4	(10)	9.50 (.41)	5.90 (.49)	(10)	10.82 (.45)	7.13 (.71)	(1[illegible]	10.35 (.85)	7.19 (1.59)
6	(8)	13.24 (.76)	7.77 (.34)	(2)	15.25 (.60)	10.47 (1.48)	([illegible]	15.72 (1.12)	11.08 (.95)
	(4.08)[c]			(3.33)[c]			(3.[illegible]		
Mean		9.93 (2.91)	6.14 (1.44)		9.53 (2.50)	6.32 (1.83)		10.21 (3.27)	7.19 (2.41)
Correlation		.971	.960		.951	.879		.9[illegible]9	.838
Slope		1.62	.80		2.00	1.34		2.08	1.35
Intercept		3.31	2.90		2.95	1.96		2.58	2.24

NOTES:

a (n) = number of instructions per format with given exit position

b standard deviations in parenth[illegible]

c mean exit position for instructions in each organizational type

was sufficient to increase the variablility in response time disproportionately for flowcharts. Lists do not have corresponding variations in shape or reading direction, but rather reflect organization in terms of the number and size of skips required across items. These factors appear not to have noticeable effects on performance.

Support for the suggestion that branching shape complicates flowchart performance comes from comparing the slope values between organizational types. The slopes for flowcharts in the two branching categories--simple groupings and "not" groupings--are, respectively, 1.34 and 1.35 seconds. These slopes are considerably steeper than the slope for strings--.80 seconds. Thus the branching flowcharts take longer to negotiate per category than the linear. The list, on the other hand, shows only minor differences in slope between organizational types. The implication for the performance model is that the "m" or "r" parameters vary not only with the type of algorithm but also with the shape of the flowchart algorithm.

There is a final variation in the breakdown correlations in Table 5 that is systematically explainable: the correlations in the flowchart are higher for strings (r = .960) than for the two types of groupings. The correlations in the list are virtually the same across organizational types. A plausible explanation for this phenomenon lies again in the shape of the flowchart. Specifically, there are shape differences within organizational types as well as between them, and the sharpest differences are located in the two types of groupings. The exit position data presented in Tables 4 and 5 are collapsed over four- vs. six-category instructions. This distinction governs different algorithm templates, however. In strings the difference between templates is relatively minimal. The six-category template is simply an extension down the screen of the four-category template (See Figure 5). On the other hand, in both types of groupings, the six-category template is qualitatively different in shape from the four-category.

In "not" groupings the six-category template is branched while the four-category is linear (Figure 5).* Based on the previously suggested effects of branching in flowcharts, we can infer an increase in difficulty between four-category and six-category instructions that is unparalleled in the other organizational types. This increase could have produced the greater variablity observed in the response times for "not" groupings (r = .838) than in those for the other organizational types.

*The radical shift in shape is unique to this organizational type: The linearity of the shorter template reflects the logical consequences of organizing four categories into three groups around a negated compound.

In simple groupings, while the four- and six-category flowcharts are both branched, the six-category has a crossover arrow, peculiar to this template (see Figure 6). The crossover arrow runs vertically, unlike the rest of the arrows in this template (and the arrows in the other templates in the stimulus set), which run horizontally or diagonally. The crossover arrow could elicit confusion or a shift in processing, adding unpredicted increments of processing time to the six-category instructions. These increments would account for the increased variability in response time for simple groupings ($r = .879$) relative to strings.

We have seen that hierachical complexity affects task complexity in the flowchart. However, this effect is much smaller than the effect of hierarchy in prose. Furthermore, it surfaces primarily in response time and not errors. It is safe to say that while algorithms do not entirely eliminate organizational complexities, they reduce them to straightforward departures from linearity in the user's path. This spatial complexity is far less taxing than complexity in the logical computations required by prose.

Finally, we can infer that algorithms dissolve the complexities of disjunction and negation, reducing these variables to differences in the assignment of labels to route contingencies. Labeling variables in the algorithm do not appear to change the task demands.

<u>Summary</u>

We can recapitulate the results as follows: The model that predicts performance time with algorithms as a linear function of the number of categories in a reader-relevant route through an instruction, underlies most of the observed results with lists and flowcharts. The model that predicts performance time with prose as a function of length and logic parameters, and errors as a function of logic, underlies most of the variations in the prose results. Together, these models explain the principal format relations found in this study: the superiority of algorithms to prose on selected combinations of exit position, logical structure, and performance measures.

2. Unpracticed Condition

The fourth question addressed in this study was how prose vs. algorithm formats affect performance at first exposures. The results from the main condition indicated how format influences performance <u>after</u> warm-up and practice. On practical grounds, the effects of first-exposures are equally important. The aim of the unpracticed condition was to assess these effects.

All structures in this condition were logically simple, four category "and" strings with no negation. The variables were format type, trial order, and format order. Trial order was a within-subjects factor with four levels, correspo ding to the four trial positions in a block; format order was a be..een-subjects factor with three levels, based on which format occupied the first block. Format order is important here since first-block presentation constitutes the severest test of a format. The first block in the short condition was the subject's first exposure to four-category instructions and to the particular mode of presentation.* The trial order dimension tested the effect of degree of exposure to a particular format independently of block order.

The response time data are displayed in Figure 12. The functions plotted relate type of format to the trial order variable for each of the three format orders--prose first, list first, and flowchart first. (There is one graph for each order.) Inspection of the three graphs shows a striking difference between prose and algorithms. Response times for prose remain relatively uniform over trials from the first to the fourth exposure. Flowcharts and lists, however, have unprecedentedly high first-exposure times, followed by rapidly decreasing response times toward the fourth trial. By comparison with the main condition, the first-exposure time for flowcharts and lists range between 15 and 24 seconds, or three and four times the norms for those formats in the main condition. Response times for prose on all four trials are very close to those found for four-category "and" strings in the main condition.

The response time curves for algorithms have essentially the same shape in all three format orders. This implies that it was not a general lack of warm-up or familiarity with the task that led to the high early latencies for algorithms. Nor did experience with one kind of algorithm serve to level the curve for the next one. Rather, the delays appear to have arisen from the nature of the individual algorithm formats. We can infer that each algorithm format was unfamiliar, unexpected, or both, requiring a short warm-up of two or three trials to elicit normal speeds.** Prose, on the other hand, was the expected and familiar medium for instructions and appeared to require little warm-up.

---

*Recall that during the introductory instructions to the experiment, subjects were shown the response buttons, were allowed to try them out, and were presented two one-category prose sentences to illustrate the response procedure.

**Looking at distinctions among the three graphs in Figure 12 reveals a possible global warm-up effect for list algorithms. The list format elicited considerably longer first and second trials when it was the first format to be encountered (Figure 12b vs. 12a and 12c). When preceded by prose and/or flowcharts, however, the first-trial delay was reduced by several seconds.

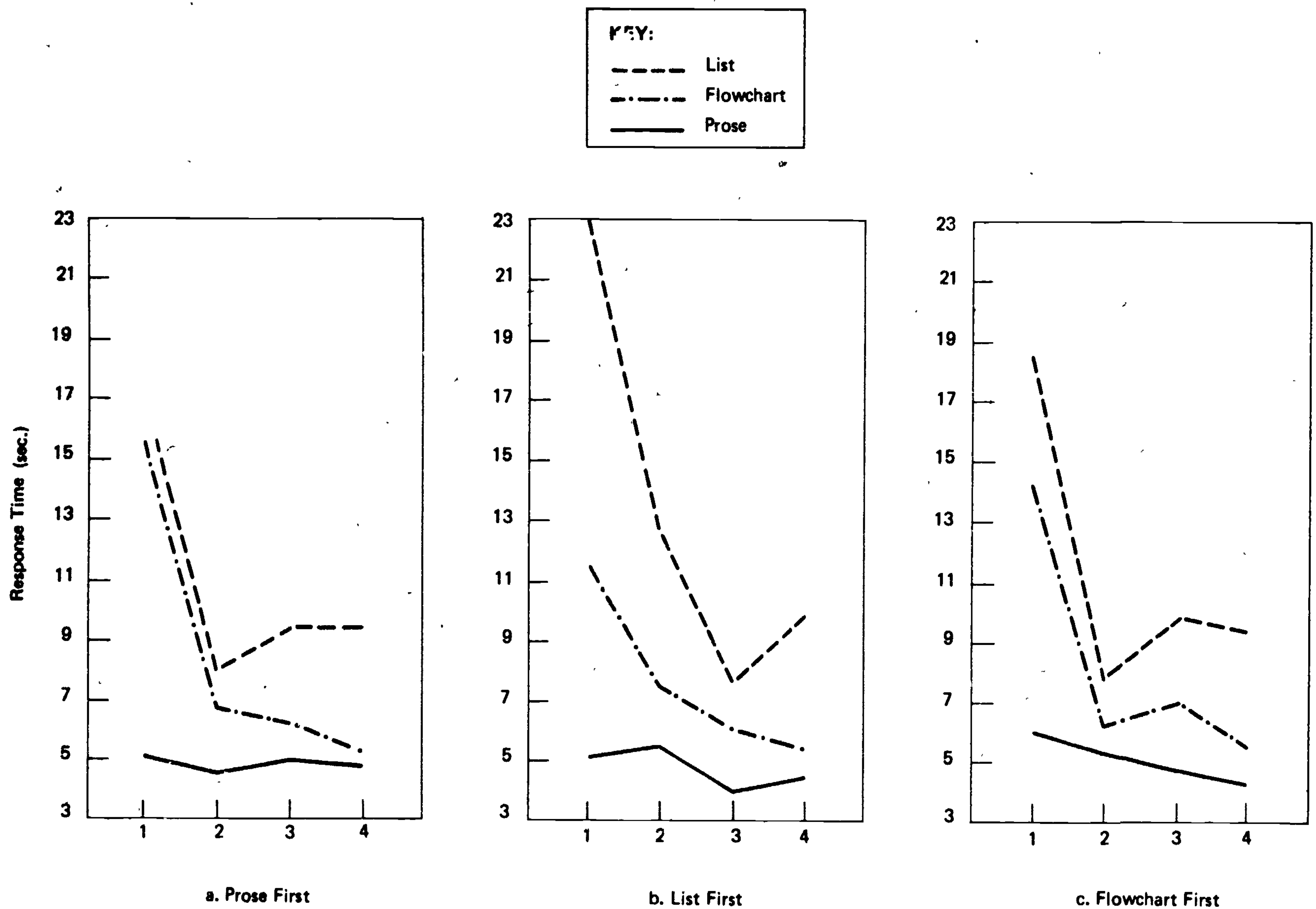


FIGURE 12. Response Time for Each Format in Unpracticed Condition as a Function of Trial Order in Each Block.

The shapes of the response time curves provide other clues to what went on in the processing of algorithms "cold." We predicted that the differences in exit position built into each block would affect performance with algorithms but not with prose. (Each block had a different distribution of one two-category and three four-category exits, as shown in Table 3.) Our predictions held for lists and prose but not for flowcharts:

Lists. The response time functions for lists (Figure 12a-c) have dips perfectly matched to the location of the two-category exit in each block. In Figure 12b, the dip corresponds to the third trial. In Figures 12a and c, the dips correspond to the second trial. The data imply that subjects were performing the list algorithm according to design, following the partial processing route specified in those instructions with early exits.

Prose. As we further expected, performance with prose did not vary with exit position. The gradual descent in response time over trials suggests that subjects were fully processing all structures.

Flowcharts. For the flowchart algorithm, however, the shape of the response time functions are not what exit position would predict. Instead of drops in response time parallel to those observed with lists, we find the gradual "learning curve" associated with prose. This result suggests that subjects in the unpracticed condition systematically misinterpreted the flowchart instructions. They may not have followed the partial processing routes specified in the design. This inference agrees with some subjects' early comments that they did not know what to do with the flowchart.

The error data further support the inference that subjects systematically misinterpreted the flowchart, but did so only in the condition where flowcharts were presented first. These data are plotted in Figure 13, relating format to trial number in terms of percent correct responses. It is clear that the systematic errors occurred in the flowchart-first condition (Figure 13c). The accuracy function for flowcharts in that condition declines dramatically on the last two trials in the block. The decline is from an average of 90% correct on the first two trials to only 20% correct on the last two. No other format in any of the three block orders shows accuracy rates below 70%.

On carefully analyzing the flowchart task, and reviewing subjects' comments and questions during the unpracticed condition, we were able to reconstruct a plausible processing strategy which appropriately violates the rules for following flowcharts and yields the error pattern in Figure 13c.

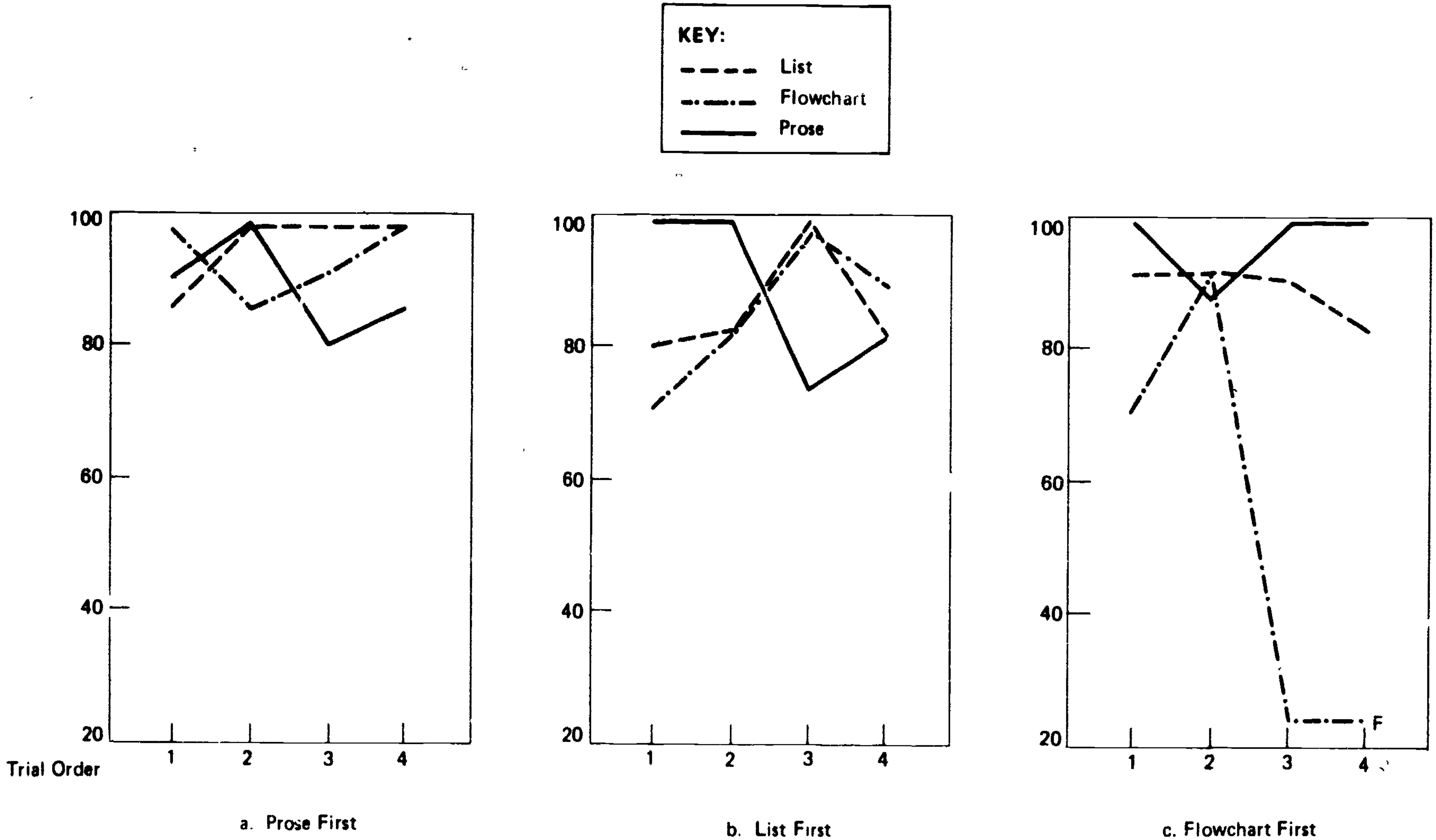


FIGURE 13. Percent Correct for each Format in the Unpracticed Condition as a Function of Trial Order.

Suppose that subjects decided to stop after the first question box in the flowchart and respond on the basis of their answer to that question. This strategy specifies: If your answer to the first question, in the first box of the flowchart, is "no," press the key specified (e.g., *); if it is "yes," press the other key, not specified (e.g., $). This strategy would have produced correct responses to all the "true" structures (since true "and" strings have "yes" for each category), and incorrect responses to any "false" structure that has "no" at some category after the first. In the truth value patterns governing Block 1, the last two trials were false: one trial had a "no" at the second category, one had "no" at the fourth (Table 3). Thus the exit-after-one strategy predicts that the last two trials in this block will receive apparently incorrect responses, while the first two trials are apparently correct. The error data for Block 1 flowcharts conform precisely to this prediction (Figure 13c).

What happened when flowcharts were presented in Block 2 or Block 3? There the error data do not imply systematic misprocessing while the time data do.

Let us suppose that subjects who received flowcharts first tended to seize on the first method of response that occurred to them, since the flowchart itself does not tell the user what to do. In this case, subjects may have been influenced by the sample sentences with one-category conditions, used to demonstrate the response procedure in the introduction to the experiment. Subjects therefore tried to respond on the basis of the first category in the flowchart.

Subjects who got flowcharts later were learning the correct use of this format as they went. Having to deal with full, four-category instructions in list form or prose form or both led subjects to expect to have to deal with the whole instruction most of the time. These subjects processed flowcharts according to design, fully processing late-exit instructions. Because these subject were learning as they went, they took longer on the two-category exit in trial 2 (where it fell in Blocks 2 and 3), but they handled it correctly. Building up to speed, they handled the remaining two trials, with four-category exits, relatively more quickly.

Why are flowcharts subject to misinterpretation while lists are not? Lists are verbally explicit. The commands for traversing a particular list algorithm are the commands for how to follow the list generally. Flowcharts are verbally inexplicit, substituting arrows for written directions. This may confuse

readers who expect to be told rather than shown what to do.*

A final aspect of the data in Figures 12 and 13 concerns the nature of the logical structure tested in the short condition: How do prose and algorithms compare for the easiest logic, as represented in the all-affirmative "and" strings?

First, the prose strings in the unpracticed condition were found to be highly accurate and nearly equivalent on the whole to algorithms (with the exception of the flowchart-first condition). It appears that when the logic is at the simplest levels, flowcharts do not save the user any errors.

The results for response time are somewhat more complicated. We will use the last, most practiced trial in each block as the point of comparison. This trial had the exit position at the fourth and last category. The critical comparison is between prose and flowcharts (lists were considerably slower than late-exit prose "and" strings in the main condition). On the last trial we find that prose is consistently between 0.5 and 1.0 seconds faster than flowcharts. Thus it appears that when the logic is straightforward and full processing is necessary, prose works quickest.

This result is consistent with our hypothesis that the flowchart may slow the reader down if rule extraction is easy. It does this by making extra physical demands: requiring separate consideration of individual categories, extra encoding operations (the "yesses" and "noes"), and longer eye movements between categories. The only circumstances in which we would expect the algorithm to save time for simple logic is when the structure to be formatted has a high partial-processing potential.

Conclusion. The unpracticed condition showed that subjects' format-specific expectations and experience interact powerfully with the performance demands of each format isolated in a task analysis. The analysis that predicts a performance advantage for algorithms considers an ideal user. For real users, algorithms did not fare so well next to prose when presented "cold." On first- and second-exposures, algorithms took several times as long as prose and were no more accurate. They also took several times as long as the algorithms for corresponding structures in the main condition, where practice preceded measured performance.

*There is a further, ad hoc factor which may add to the explanation. The flowcharts in this experiment may have been less than optimally presented because the computer display constraints prohibited our expanding the layout or using tips on arrows. Both changes would have perhaps clarified directional links between boxes.

On the other hand, prose presented cold gave no problems. Prose was almost equally fast and accurate across trials in the unpracticed condition. Furthermore, the scores on those trials were representative of the post-practice scores on similar structures.

We can infer that algorithms are unexpected or unfamilar, creating initial delay and strong warm-up effects. Prose instructions, on the other hand, are expected and familiar, and needed little warm-up for users to reach optimal levels of accuracy and speed. We can also infer that the effect of the novel formats is early and temporary, reduced by experience over trials.

Flowchart algorithms were not just unfamiliar. They also appeared to be inscrutable to subjects at first glance. Flowcharts were systematically misinterpreted when they were presented first. The procedure for lists, on the other hand, was much easier to figure out and lists were highly accurate from the beginning.

For the designer who is considering algorithm formats, the results of the short condition suggest several caveats. First, the typical reader presented with flowcharts may need brief directions on how to use them. Second, the reader presented with either lists or flowcharts may need some sort of introductory statement to prepare the reader for a novel format. (This may help reduce the reader's initial surprise and delay.) Third, for the easiest logic (affirmative "and" strings) algorithms in either list or flowchart form probably won't help performance. On the contrary, they may slow it down; this will depend on the length of the conditional clause and the likelihood of early exits.

3. Preference Data

Our final question concerned how well readers would like each format. At the end of the experiment, we asked subjects to rank the three formats in order of preference. As shown in Table 6, the preference for algorithms was overwhelming. Flowcharts were ranked first two and a half times as often as lists; and lists, three times as often as prose. Thus preference followed measured difficulty.

Table 6. Distribution of Formats by Subjects' Preference Rankings.

RANK ORDER	FORMAT Flowchart	List	Prose	Total
1	35	14	5	54
2	16	35	3	54
3	3	5	46	54

What independent variables or subject characteristics correlate with order of preferred formats? First, consider some potentially influential variables: For subject characteristics, some likely correlates of preference are sex of subject, degree of experience with programming or logic, and level of education.

For experimental variables, likely correlates are the order of presentation of the three formats and the difficulty of the three organizational types (the between-subjects factor). Conceivably, subjects receiving the harder structures would tend more than other subjects to prefer algorithms over prose.

To look at the relationship between these variables and format preference, we constructed tables showing the distribution of each variable by first-ranked format. Because the data displayed in each table were reasonably clear, we did not perform statistical tests of the relationships. In brief, the levels of a given variable appeared nearly equivalent in their distributions with respect to highest-ranked format. Thus, the five variables seem to have little to do with format preference.* Indeed, the fact that neither easy vs. hard prose nor order of format presentation affected format preference, attests to the durability of the obtained ordering.

*Although our attempt to tap prior experience with algorithms and with complex logic failed to reveal a reliable relationship to preference, there were problems with our probe questions ("Have you had a programming or logic course?"): (1) We conflated experience in using diagrams (which should favor flowcharts) with practice in truth-functional logic (which should help prose). (2) We did not specify the kind and amount of this experience. More appropriately probed, this kind of experience may demonstrate links to format preferences.

The only apparently significant factor in first format preference was a final variable: the nature of subjects' academic or professional field (for students and AIR employees respectively). This factor distinguished flowchart from list preferences particularly. The distribution of subject's field by first format preference is shown in Table 7.

Table 7. First Format Preference by Nature of Subject's Academic or Professional Field.

	FIRST-RANKED FORMAT		
FIELD	Flowchart	List	Prose
Liberal Arts/Ed.	9	8	1
Sciences	17	3	4
Public	9	3	-

KEY

Liberal Arts/Ed.: English, French, music, art, education

Sciences: linguistics, economics, psychology, political science, mathematics, sociology

Public: law, public administration, international relations, business

The table also shows how specific fields were classified into three general areas: liberal arts (e.g. English, music), science (e.g., economics, linguistics, math), and public areas (e.g., law, business). It is clear that most people in science fields preferred flowcharts to lists, while people in liberal arts and education were split between preferring lists and preferrng flowcharts. The difference makes sense if we consider the thinking style typical of people in sciences vs. liberal arts: the former tend to have a spatial/math orientation, the latter a verbal orientation.

Conclusion

There are two important limitations on conclusions drawn from the preference data. First, if attitudes had been queried at the beginning of the experiment, before subjects had experienced how algorithms helped performance in the long run, subjects might have

preferred prose. Second, the laboratory context may have created acceptable formats artificially. That possibility suggests a problem with using experimental performance data as the basis for format recommendations. More realistically, some algorithm advocates (e.g., Wason & Johnson-Laird, 1972) have warned that readers in everyday contexts may initially balk at the algorithm despite its being cognitively simpler. This may be because algorithms are new or formidable looking. Or it may be because they deny the context, choice, and rationale that some readers desire.

Considering these limitations, our data allow this conclusion: when readers realize how algorithms facilitate logically difficult decision-making, they like algorithms better than prose. When given algorithms, readers prefer flowcharts to lists.

V. SUMMARY AND CONCLUSION

Conditional instructions were easier as algorithms than as prose when subjects were prepared for algorithms by practice with feedback. In general, the algorithm helped performance when the condition in the instruction was logically complex. The more complex the condition, the more algorithms helped. We explained this result in terms of a processing model in which performance with algorithms stays relatively constant across changes in logical structure, while prose performance declines as the logic becomes more difficult.

Prose

The components of logical difficulty in the prose instructions included disjunction ("or"), negation (simple and compound), and hierarchical grouping of categories. Prose performance declined as the organization of categories in the condition went from linear to simple groupings and from simple groupings to grouping through compound negation. Similarly, within organizational types, performance declined from "and" connectives to "or" connectives. The inherent difficulty of "or" was exacerbated by commas in an implicit "or" series. Negation of single categories seemed to cause somewhat slighter decrements than the other variables.

The decrements appeared in both speed and accuracy. The difference between the easiest and the hardest prose structures (averaged over four and six categories) was between five and six seconds in response time and about 33% in errors.

Algorithms

By contrast, algorithms were highly accurate, and their speed varied little, across organizational types, across "and" and "or" structures, and across affirmative and negative structures. Except for a slight increase in difficulty created by branching flowcharts (representing hierarchical organization), algorithms virtually eliminated the logical processing asymmetries in prose.

Predictably, the maximum improvement with algorithms was at the hardest level of prose. The flowchart gained a six-second savings in time, and a 33% savings in errors for the most difficult prose structures (collapsing over length). For other difficult structures, the savings were a little over four seconds and 12% for the average grouping, two seconds and 12% for the "or" strings. There were no appreciable savings for "and" strings.

Reinforcing the advantage of algorithms in performance, an overwhelming majority of subjects indicated at the end of the study that they liked algorithms better than prose.

The overall superiority of algorithms was qualified, however, by three complications revealed in the performance data:

- The difference between flowcharts and lists, which changed the speed relationships between prose and algorithms.

- The effect of partial processing, which also changed the speed relationships.

- The effect of first exposure to the instructions, which drastically reversed the algorithm advantage in both speed and accuracy.

Let us briefly review each of these complications.

Flowcharts vs. lists. Both forms of algorithms were equally accurate. Lists, however, were consistently slower than flowcharts, by about three-quarters of a second for each relevant category in the condition. This means that for six-category late-exit conditions, the difference in response time between the two formats was about 4.5 seconds.

We explained this difference using a model of the physical task entailed by each format. In this model, lists require extra encoding, scanning, and matching operations to follow the "go to" commands. Flowcharts eliminate these operations by marking the route visually with arrows and by repeating sequences of contingent questions and placing them near each other.

Because lists are slow, averaging lists and flowcharts narrows the difference in speed between algorithms and prose for the difficult structures. For the easier, string structures, lists were actually slower than prose for both "and" and "or" connectives.

Supporting the performance data, we found that subjects preferred flowcharts to lists by a ratio of about five to two.

Partial processing. The second significant complication concerned the exit position variable, which governs the partial processing potential of the instructions. The data we have summarized to this point were collapsed over exits. When we examined the exit breakdowns, we found the response time relations altered between prose and algorithms. Specifically, while exit position made no systematic difference for prose, it exerted dramatic effects on the speed of responding to algorithms:

Response time was a linear function of the exit position in the instruction.

. These effects were explained in the model we developed of the processing task required in each format. To capture the exit data, the model (1) assumed exhaustive processing for prose and (2) posited constant increments of response time for the list and the flowchart, added for each reader-relevant category in the basic instruction.

The implications for the prose-algorithm comparison are obvious. Exit position makes no difference for the relative accuracy of the formats. But for response time, the advantage of algorithms decreases the more there is to be processed in an instruction (late exits); it increases the more irrelevant content occurs (early exits).

The consequences of this variation were most significant in the case of the easy "and" strings. The collapsed data suggested that these strings should not benefit from flowchart translation, since flowcharts appeared to save neither errors nor time. However, flowcharts turned out to be distinctly better than prose when the exit was early--gaining 3.5 seconds in response time (averaged over four and six categories).

First exposures. The third source of complications in the data was the degree of the subject's task-specific experience with algorithms. That variable was operationalized as experience over the course of the experiment and tested by practiced vs. unpracticed performance. The data summarized so far were based on performance after practice. When we looked at unpracticed performance, we found that algorithms were no longer faster and more accurate than prose. Lists and flowcharts took from three to five times as long as prose on the first and second trials of the unpracticed condition--subjects' first exposures to the instructions. Latencies for both forms of the algorithm then declined dramatically over trials, while prose was uniformly fast (with a slight learning curve) from the beginning. Furthermore, subjects who saw flowcharts first in the short condition systematically misused them. Accuracy levels reached lows of 20%, while lists and prose were consistently above 80%.

Implications for the processing of formats are fairly clear. Prose is the expected and traditional mode of presenting instructions and requires little explanation or warm-up. Lists and flowcharts, on the other hand, are unexpected and unfamiliar and require two or three practice readings before they can be used efficiently. Moreover, while flowcharts elicit the best performance after practice, they appear the least explicit in telling the user what to do. The list algorithm, more like prose, is much more transparent in directing the user.

* * * * * * * * * * *

In conclusion, this study confirmed our basic expectations about the relative efficiency and effectiveness of prose and algorithms. The results thus validate our task analyses and support the theoretical models developed for processing complex conditional instructions in prose and algorithm formats.

Recommendations

The results of this study translate into recommendations to document designers about where in the set of conditional instructions to apply algorithms and under what circumstances. The recommendations below pertain to the structures used in the current stimulus set.

In general, we would recommend using the flowchart algorithm to reduce the time and errors involved in complex conditional instructions. If the audience for the instructions consists of average readers with non-technical backgrounds, then the flowchart should be prefaced with brief directions on how to use it. Effective directions might be quite simple, for example: "You are going to see a set of boxes and arrows. Read the questions in the boxes and, depending on your answer, follow the arrow to the next box or the next command." (We did not test the effectiveness of instructions like this on unpracticed readers.)

The list algorithm is also a good way to simplify instructions, although it is somewhat slower than the flowchart. Because it is less confusing, the list needs less preface than the flowchart for the average reader. A simple preparatory statement (e.g., "You are going to see instructions in a new form.") may help curb the surprise and delay that comes with first exposures.

More specifically, the place to use algorithms in a set of complex conditional instructions is where the condition on the instruction has a difficult structure. A difficult structure in our stimulus set means:

- any grouping of categories
- any string with "ors"
- any structure with compound negation ("not both/and" and "neither/nor").

One the other hand, it is usually not necessary to use an algorithm if the structure is easy. Easy in our stimulus set means

- "and" strings with all affirmative categories

- "and" strings with two negated categories.

In one circumstance, however, algorithms can improve the speed of responding to conditions with easy structures. This is when the condition does not require full processing. The partial processing potential of the condition depends on the structure of the condition and the distribution of truth value characteristics in the audience (for reader-referenced instructions such as used in this experiment). When information about the relevant truth characteristics is available, then designers can determine whether early exits are likely to occur and decide on that basis whether to use algorithms to reduce response time.

Practical Considerations

These recommendations have stayed close to the experimental data. Looking toward real-world applications, we find a variety of situational and stimulus factors that limit the generality of our recommendations. Let us consider the more salient of these factors and how they might qualify our evaluation of algorithm formats.

The nature of the stimulus instructions. Given the domain of truth-functional conditions on instructions, we have not considered how algorithms will affect conditions longer than those we investigated, or conditions with more internally involved categories--such as "if you are male and have a yearly income greater than one-third the amount of your annual housing costs," rather than "if you are male and married." Nor have we considered conditions in which the categories are external tests that require physical actions--i.e., procedures. All these variables could potentially influence how algorithms work. We will focus on how our findings extend to longer instructions, since the length variable allows some straightforward predictions.

For instructions longer than six categories, algorithms should have an even stronger advantage than found in this study, for two reasons: (1) When the condition or an instruction grows longer than six or seven categories, it is beyond the span of working memory. Prose processing should break down, as may be manifested in errors, back-referencing, starting over, or resorting to outside aids. The processing of algorithms, on the other hand, should stay uniformly easy since algorithms place negligible memory demands on the user. (2) Longer conditions

should profit more from the partial processing imposed by the algorithm, given a high probability of early exits.*

The nature of the task. Two primary functions of practical reading are learning and doing. The conclusions from this study are probably limited to doing. If the task is, for example, learning a rule, algorithms probably preclude understanding and remembering. (The interaction of format with function is discussed more fully in Holland, forthcoming.)

In what contexts is doing the primary function? It appears central in public documents such as forms. For example, a benefits applicant need not understand the structure and rationale behind a sequence of contingent questions in order to answer correctly and bring about a fair determination of eligibility.** Doing is also primary in one-shot instructions and procedures, intended for immediate performance rather than for transmission and retention of information.

Variations in the visual and presentational form of each format. We discussed this issue earlier, noting that the constraints of computer display in this study imposed a flowchart layout which may have been unnecessarily confusing. Similarly, the display medium may have prevented the list format from being used as efficiently as possible. Being able to point to and trace the specified route with a finger, as on paper, might have mitigated the demands of visual leaps in the computerized list.

Though it is clear that visual layout and presentational medium affect performance generally, very little is known specifically about the effects of graphical and display characteristics on processing, or about which characteristics are relevant for deciding the optimal form of a given format. However, if we assume that the algorithm formats in this experiment were less than optimally presented, then we can infer that better, less constrained presentations would not merely enhance the algorithm advantage but might also reduce the procedural ambiguity of the flowchart evident at first exposures.

*Unlike the flowchart, the list algorithm should increase in response time with increasing length, despite a fixed exit position. This is because the list requires scanning from the exit question to the end of the entire set of questions to reach the outcome commands.

**However, for noncategorical questions (e.g., "Why do you want this job?" vs. "Are you male?"), indications are that readers need to understand the rationale to successfully complete the form (Holland & Redish, 1981).

Functional criteria for evaluating differences among formats. We have reported time and error differences without judging their practical significance. How critical are effects of the magnitude observed here in deciding among alternative formats? This will depend on the function and context of the instructions.

For example, we found time savings of two to six seconds gained by flowcharts. Such savings are frequently of little practical significance to users and designers of instructions. Seconds may make a difference, however, for emergency procedures under time-limited conditions. Also, for forms with very large audiences, seconds translate into enormous amounts of time. Just consider how many person-hours are saved if 20 million people could save 10 seconds each when filling out the IRS Form 1040.

Savings in errors are more clearly important across contexts. Error savings ranged from 10% to 33% on the instructions for which we have recommended algorithms. Errors of this magnitude on forms and other public documents have been reported to have costly consequences for users and institutions (Redish, 1979; Battison, 1980; Charrow, Holland, Peck & Shelton, 1980).

Effects on response time should become similarly decisive across contexts when they begin to approach magnitudes of a minute. Users experiencing such delays might be reluctant to complete an instruction or to approach other instructions, given that users' motivation to finish is not intrinsically high. Delays of up to 25 seconds occurred in our data at the point of first exposures to algorithms.

Another functional criterion in evaluating performance data is design constraints. Final choices about what format to use require weighing the expected gains in time and accuracy against the costs of designing and producing algorithms. Algorithms usually need more space on the page and require more preparation time. Not only does prose take up less space, but writing out instructions in prose is generally just the first step in constructing the algorithm.

Differences in attitudes and acceptance. As we observed earlier, people's attitudes toward novel formats may vary between experimental and real-world settings, and among different individuals and populations of readers. Attitude in turn affects whether readers will try out and continue to use novel formats, as well as how quick and accurate readers will be. Our experimental finding that people prefer algorithms to prose may be context-bound. The scant evidence on attitudes toward and acceptance of algorithms in non-experimental settings is equivocal.

Some observations suggest that even practiced algorithm users may prefer prose because it allows them to control the reading process and to understand the rule underlying the instruction (Wason & Johnson-Laird, 1972). It has also been observed that when algorithms are applied to the automatic translation of specialized skills, the specialists sometimes resent it. Professionals may perceive such applications as an attempt to reduce and publicize what they consider complex and privileged knowledge. (See the review of algorithms in medical and legal practice in Holland, forthcoming.) Finally, some audiences, particularly those without education or experience in using charts and graphs, seem to be afraid of technical-looking formats and simply omit the instructions or avoid the document when such options are possible (Holland, 1980; Charrow, Holland, Peck, & Shelton, 1980).

However, two studies of documents that were revised to include algorithms or aspects of algorithms (Kammann, 1975; Charrow, Holland, Peck & Shelton, 1980) found that the intended audience accepted, used, and preferred the revised material. These audiences included people without a technical background.

Field testing of new formats with target audiences may be the only way to predict performance and utilization at this point in our knowledge.

Degree of experience and ability. The results of this study suggest that flowchart procedures can be confusing, if not inscrutable, at first encounter. The novice needs assistance--practice with feedback or explicit instruction. We suggest that extremely short-term assistance can be effective. Most of our subjects, professionals or graduate and undergraduate college students, were puzzled only initially and grasped the correct procedure early on. We can assume that most were good readers and had had some experience with charts and graphs, and that these factors contributed to their quick understanding. It is worth noting, however, that a few individuals continued to misuse the flowchart throughout the experiment. Their accuracy levels with flowcharts were only a little above chance and far below lists and prose. Only five of the 54 subjects fell into this category, and nothing in the data we gathered on subjects systematically distinguished them. But we can infer that there may be some populations in which readers will have a great deal of difficulty with the flowchart form of the algorithm.

We do not know what factors are critical to predicting the individual's ability to follow graphical formats: How much previous education or experience and of what specificity are prerequisite to interpreting the symbols in flowcharts? What individual differences in spatial abilities or cognitive style affect the capacity to handle flowcharts or lists with ease?

* * * * * * * * * *

Clearly there are a range of stimulus, audience, and task variables to consider in applying the recommendations of this study to particular problems in presenting instructions. We have covered only some of the relevant variables. Even if we could specify all the relevant influences in a particular case, we have no theory which would allow us to predict from that specification the effectiveness of competing formats. The gaps in empirical and theoretical knowledge suggest that when the designer considering algorithms is uncertain of the impact of contextual factors, that designer would do well to pilot test algorithms in the intended context. The pilot results could then be judged within the larger context of the relative costs and gains of adopting innovative formats.

REFERENCES

Battison, R.B. Document design: Language planning for paperwork. Language Planning Newsletter (Honolulu, Hawaii), November, 1980.

Blaiwes, A. Formats for presenting procedural instructions. Journal of Applied Psychology, 1974, 59, 683-697.

Braine, M.D.S. On the relation between the natural logic of reasoning and standard logic. Psychological Review, 1978, 85, 1-21.

Charrow, V.R., Holland, V.M., Peck. D.G., & Shelton, L.V. Revising a Medicaid recertification form: A case study in the document design process. Washington, D.C.: American Institutes for Research, 1980.

Davies, I. K. Get immediate relief with an algorithm. Psychology Today, 1970, 3, 53.

Follettie, J.F. Numerical display research: Review and discussion. (Technical Report 2-78-01). Los Alamitos, Calif.: SWRL Educational Research and Development, 1978.

Follettie, J.F. Graph-using profiencies of upper elementary schoolers. (Technical Report 63). Los Alamitos, Calif.: SWRL Educational Research and Development, 1979.

Gane, G. P., Horabin, I.S. & Lewis, B. N. The simplification and avoidance of instruction. Industrial Training International, I. No. 4, 160-6, 1966.

Holland, V.M. Revising a government document: The case of the Medicaid recertification form. Paper presented at the Symposium on Writing and Designing Documents: Research and Practical Solutions. Carnegie-Mellon University, October, 1980.

Holland, V.M. Algorithms for presenting instructions: A review of the literature. Washington, D.C.: American Institutes for Research, forthcoming.

Holland, V.M. & Redish, J.C. Strategies for understanding forms and other public documents. To be published in D. Tannen (Ed.), Proceedings of the Georgetown Round Table on Discourse: Text and Talk.

Holland, V.M. & Rose, A.M. Understanding instructions with complex conditions. (Technical Report No. 5). Washington, D.C.: American Institutes for Research, 1980.

Jones, S. Design of instruction. London: Her Majesty's Stationery Office, 1968.

Kammann, R. The comprehensibility of printed instructions and the flowchart alternative. Human Factors, 1975, 17, 90-113.

Landa, L. Instructional regulation and control: Cybernetics, algorithmization, and heuristics in education. Englewood Cliffs, N.J.: Educational Technology, 1976.

Osherson, D.N. Logic and models of logical thinking. In R.J. Flamagne (Ed.), Reasoning: Representation and processing children and adults. Hillsdale, N.J.: Erlbaum, 1975.

Quine, W.V.O. Methods of logic. New York, N.Y.: Holt, Rinehart, & Winston, 1972.

Redish, J.C. The Language of the Bureaucracy. (Technical Report No. 15). Washington, D.C.: American Institutes for Research, 1980.

Rose, A.M. & Cox, A. Following instructions. (Technical report). Washington, D.C.: American Institutes for Research, 1980.

Wason, P.C. The drafting of rules. The New Law Journal, 1968, 118, 548-549.

Wason, P.C. & Johnson-Laird, P.N. Psychology of reasoning: Structure and content. Cambridge, Mass.: Harvard University Press, 1975.

Wheatley, P. & Unwin, D. The algorithm writer's guide. London: Longman's, 1972.

Wright, P. Decision-making as a factor in the ease of using numerical tables. Ergonomics, 1977, 20, 91-96.

Wright, P. & Reid, F. Written information: Some alternatives to prose for expressing the outcomes of complex contingencies. Journal of Applied Psychology, 1973, 57, 160-166.

Appendix A

ANOVA Tables for Each
of the Three Types of Organization

Strings Response Time*

	SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F	TAIL PROBABILITY
1	Mean	156.89996	1	156.89996	1361.04	.000
	Error	1.95975	17	.11528		
2	Format (F)	1.64117	2	.82058	46.87	.000
	Error	.59529	34	.01751		
3	And-Or (A)	.04055	1	.04055	6.40	.022
	Error	.10775	17	.00634		
4	FA	.04738	2	.02369	4.29	.022
	Error	.18768	34	.00552		
5	Length (L)	.45311	1	.45311	223.93	.000
	Error	.03440	17	.00202		
6	LF	.05746	2	.02873	12.51	.000
	Error	.07806	34	.00230		
7	AL	.00002	1	.00002	.02	.885
	Error	.01687	17	.00099		
8	LAF	.00025	2	.00013	.07	.930
	Error	.05919	34	.00174		

*The log transforms of response time data were used for the ANOVA's. Order of format blocks was [illegible] in the response time analyses.

Strings Number Correct

	SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F	TAIL PROBABILITY
1	Mean	7004.16699	1	7004.16699	2980.50	.000
	Order (O)	.25000	2	.12500	.05	.948
	Error	35.24998	15	2.35000		
2	Format (F)	7.19444	2	3.59722	4.25	.024
	FO	5.22222	4	1.30555	1.54	.216
	Error	25.41666	30	.84722		
3	And-Or (A)	.66667	1	.66667	2.08	.170
	AO	2.19445	2	1.09722	3.42	.060
	Error	4.80556	15	.32037		
4	FA	3.52778	2	1.76389	5.65	.008
	FAO	4.94444	4	1.23611	3.96	.011
	Error	9.36111	30	.31204		
5	Length (L)	.00000	1	.00000	.00	1.000
	LO	.02778	2	.01389	.08	.924
	Error	2.63889	15	.17593		
6	LF	.08333	2	.04167	.29	.753
	LFO	.38889	4	.09722	.67	.619
	Error	4.36111	30	.14537		
7	AL	.16667	1	.16667	2.57	.130
	ALO	.19444	2	.09722	1.50	.255
	Error	.97222	15	.06481		
8	LAF	.08333	2	.04167	.17	.848
	LAFO	.55556	4	.13889	.55	.698
	Error	7.52778	30	.25093		

Simple Groupings Response Time

	SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F	TAIL PROBABILITY
1	Mean	173.75748	1	173.75748	1312.02	.000
	Error	2.25140	17	.13244		
2	Format (F)	2.16048	2	1.08024	46.79	.000
	Error	.78496	34	.02309		
3	And-Or (A)	.03689	1	.03689	16.43	.001
	Error	.03817	17	.00225		
4	FA	.05490	2	.02745	16.54	.000
	Error	.05641	34	.00166		
5	Length (L)	.67835	1	.67835	221.35	.000
	Error	.05210	17	.00306		
6	LF	.00864	2	.00432	1.97	.155
	Error	.07446	34	.00219		
7	AL	.00554	1	.00554	5.21	.036
	Error	.01808	17	.00106		
8	LAF	.00891	2	.00446	2.36	.110
	Error	.06431	34	.00189		

Simple Groupings Number Correct

	SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F	TAIL PROBABILITY
1	Mean	7015.55859	1	7015.55859	5951.93	.000
	Order (O)	.84259	2	.42130	.36	.705
	Error	17.68056	15	1.17870		
2	Format (F)	15.59258	2	7.79629	7.06	.003
	FO	5.46296	4	1.36574	1.24	.316
	Error	33.11108	30	1.10370		
3	And-Or (A)	1.04167	1	1.04167	4.06	.062
	AO	1.19444	2	.59722	2.33	.132
	Error	3.84722	15	.25648		
4	FA	4.33333	2	2.16667	6.03	.006
	FAO	2.05556	4	.51389	1.43	.248
	Error	10.77777	30	.35926		
5	Length (L)	.56019	1	.56019	3.34	.087
	LO	.34259	2	.17130	1.02	.384
	Error	2.51389	15	.16759		
6	LF	.59259	2	.29630	1.58	.222
	LFO	1.62963	4	.40741	2.18	.096
	Error	5.61111	30	.18704		
7	AL	.37500	1	.37500	1.53	.235
	ALO	.02778	2	.01389	.06	.945
	Error	3.68056	15	.24537		
8	LAF	.33333	2	.16667	.66	.526
	LAFO	.22222	4	.05556	.22	.926
	Error	7.61112	30	.25370		

"Not" Groupings Response Time

	SOURCE	SUM OF SQUARES	DEGRRES OF FREEDOM	MEAN SQUARE	F	TAIL PROBABILITY
1	Mean	188.76895	1	188.76895	1792.42	.000
	Error	1.79035	17	.10531		
2	Format (F)	3.20946	2	1.60473	29.75	.000
	Error	1.83381	34	.05394		
3	And-Or (A)	.00684	1	.00684	4.87	.041
	Error	.02387	17	.00140		
4	FA	.09984	2	.04992	15.26	.000
	Error	.11120	34	.00327		
5	Length (L)	1.40284	1	1.40284	387.43	.0[illegible]
	Error	.06155	17	.00362		
6	LF	.02458	2	.01229	4.62	.017
	Error	.09042	34	.00266		
7	AL	.00035	1	.00035	.14	.709
	Error	.04127	17	.00243		
8	LAF	.06597	2	.03298	12.69	.000
	Error	.08835	34	.00260		

"Not" Groupings Number Correct

	SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F	TAIL PROBABILITY
1	Mean	5683.62695	1	5683.62695	2757.56	.000
	Order (O)	5.12038	2	2.56019	1.24	.317
	Error	30.91666	15	2.06111		
2	Format (F)	94.45366	2	47.22683	29.62	.000
	FO	11.54630	4	2.88658	1.81	.153
	Error	47.83330	30	1.59444		
3	And-Or (A)	.46296	1	.46295	.97	.341
	AO	.00926	2	.00463	.01	.990
	Error	7.19445	15	.47963		
4	FA	.45370	2	.22685	.52	.599
	FAO	2.32407	4	.58102	1.34	.280
	Error	13.05555	30	.43519		
5	Length (L)	4.16667	1	4.16667	8.24	.012
	LO	.25000	2	.12500	.25	.784
	Error	7.58334	15	.50556		
6	LF	.02777	2	.01389	.02	.978
	LFO	6 97223	4	1.74306	2.83	.042
	Error	18.50000	30	.61667		
7	AL	2.66667	1	2.66667	10.99	.005
	ALO	1.02778	2	.51389	2.12	.155
	Error	3.63889	15	.24259		
8	LAF	5.58334	2	2.79167	6.31	.005
	LAFO	3.30556	4	.82639	1.87	.142
	Error	13.27779	30	.44259		

Appendix B

Summary of Mean Response Times (Sec.)

Structures (b)	ORGANIZATIONAL TYPES											
	Strings				Simple Groupings				"Not" Groupings			
	Exit (c)	P (d)	L	F	Exit	P (d)	L	F	Exit	P (d)	L	F
4AET	4	6.639	9 578	5 356	2	10 361	7.511	3 994	4	7.817	9 611	6.667
4AEF	1	7 344	4 494	3 867	2	10.894	6.900	5.417	1	12 244	5.022	3.328
4AMT	4	6.833	9 828	5 528	3	10.350	8.522	4.911	4	8 700	9.761	6 250
4AMF	3	7.028	7 444	5.483	3	9 578	8.017	4 311	3	11 622	9.200	5.778
4ALT	4	6.844	8 489	6 594	4	9 094	10.344	6.200	4	8.044	9.906	5 978
4ALF	4	6 689	9 350	6 200	4	10.367	10.750	6.444	4	11 078	9 678	6.611
4OET	1	7.683	4 533	3 561	2	9 094	6.428	4.528	1	11 767	4 922	3.978
4OEF	4	10.050	9.583	5.367	2	9.5[illegible]2	6 [illegible]28	4 156	4	12 989	9.233	5.722
4OMT	3	6 428	9 006	5.528	3	7 850	7.839	5.606	3	12.922	7.856	5.639
4OMF	4	7 522	9 844	6 011	3	8.522	8.122	5.822	4	12 106	10.000	5.789
4OLT	4	9 450	9 433	5 672	4	8.439	11.128	7 083	1	11.717	4.489	4.022
4OLF	4	7.922	9.367	6.544	4	8.928	10.083	6.728	4	16.000	10.194	5.233
6AET	6	7 128	11.928	8.283	2	14.828	7.039	4.456	4	14.389	10.550	8 217
6AEF	1	8 172	5.217	3.600	4	12.344	10.628	7.206	2	18 328	7.683	5.600
6AMT	6	7.189	13 372	7.456	4	11.744	11.194	8.500	4	14.144	10.550	10.033
6AMF	4	8.128	9 933	6 322	4	15.372	11 328	6.789	4	14.633	11.989	9.444
6ALT	6	6.972	13.772	8 228	6	15.483	15.672	11.511	6	15.656	15.778	11.817
6ALF	6	7 872	13.000	7.611	2	11 311	9.350	6.428	6	14 567	15.850	10.450
6OET	1	8.061	6 561	4.161	4	9.656	10.417	7 256	2	16.167	8.667	6.600
6OEF	6	9.383	13.400	7 639	2	15.544	6.200	4.100	4	13.817	11.067	7.606
6OMT	4	8 683	9.622	5.42[illegible]	4	9 561	10.917	6 928	3	14.744	9.183	7.022
6OMF	6	9 333	12 400	7.600	4	13.733	11.406	8 150	4	14 806	11.661	8.756
6OLT	6	10 083	14 078	7 967	2	12 350	7 922	5.806	6	13.850	17.400	10.089
6OLF	6	8.878	13 956	7.411	6	11 667	14 817	9 422	6	17.739	14.667	11.956

KEY

(a) n = 18 for all entries

(b) structures identified by variables in this order
 number of categories (4 vs 6)
 connective (A vs O for and vs or)
 exit level (E-M-L for early-middle-late)
 truth value (T, F)

(c) exit position is the exact number of categories minimally necessary for a decision

(d) P = prose, L = list, F = flowchart

APPENDIX B. Summary of Mean Response Times (Sec) [a]